

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日                      2 0 0 3 年    3 月 2 4 日  
Date of Application:

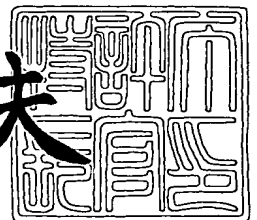
出 願 番 号                      特 願 2 0 0 3 - 0 7 9 5 5 6  
Application Number:  
[ST. 10/C] :                      [ J P 2 0 0 3 - 0 7 9 5 5 6 ]

出      願      人                      ソニー株式会社  
Applicant(s):

2 0 0 4 年    1 月    5 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 0290796108

【提出日】 平成15年 3月24日

【あて先】 特許庁長官殿

【国際特許分類】 H03M 13/03

【発明者】

    【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社  
                                内

    【氏名】 野田 誠

【発明者】

    【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社  
                                内

    【氏名】 山岸 弘幸

【特許出願人】

    【識別番号】 000002185

    【氏名又は名称】 ソニー株式会社

【代理人】

    【識別番号】 100082131

    【弁理士】

    【氏名又は名称】 稲本 義雄

    【電話番号】 03-3369-6479

【手数料の表示】

    【予納台帳番号】 032089

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

    【包括委任状番号】 9708842

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 符号化装置および符号化方法、記録媒体、並びにプログラム

【特許請求の範囲】

【請求項 1】  $m$  ビットの情報語を  $n$  ビットの符号語に連続して変換する符号化装置において、

入力された情報系列に、所定の間隔で第 1 の DC 制御ビットを挿入して、第 1 の情報系列を生成するとともに、前記所定の間隔で前記第 1 の DC 制御ビットとは異なる第 2 の DC 制御ビットを挿入して、第 2 の情報系列を生成する情報系列生成手段と、

前記情報系列生成手段により生成された前記第 1 の情報系列を、情報語長  $m$  対符号語長  $n$  の変換比率で符号変換して、第 1 の仮符号系列を生成する第 1 の符号変換手段と、

前記情報系列生成手段により生成された前記第 2 の情報系列を、情報語長  $m$  対符号語長  $n$  の変換比率で符号変換して、第 2 の仮符号系列を生成する第 2 の符号変換手段と、

前記第 1 の符号変換手段により生成された前記第 1 の仮符号系列と前記第 2 の符号変換手段により生成された前記第 2 の仮符号系列とのうち、いずれか一方を選択する選択手段と

を備え、

前記第 1 の符号変換手段および前記第 2 の符号変換手段が用いる符号化規則は、前記符号化規則を有限状態型の符号変換表で表現した場合、所定の始点状態から符号化された前記第 1 の仮符号系列の第 1 の符号状態と、前記所定の始点状態から符号化された前記第 2 の仮符号系列の第 2 の符号状態とが同一となった時点で、前記第 1 の仮符号系列を構成する符号化ビットの合計の 2 の補数と、前記第 2 の仮符号系列を構成する符号化ビットの合計の 2 の補数とが、常に異なるように、情報語に符号語が割り当てられている

ことを特徴とする符号化装置。

【請求項 2】 前記第 1 の符号変換手段および前記第 2 の符号変換手段は、前記情報語長  $m$  を 2 ビットとして、2 ビットを符号変換単位として符号変換を実

行する

ことを特徴とする請求項 1 に記載の符号化装置。

【請求項 3】 前記第 1 の符号変換手段および前記第 2 の符号変換手段が用いる前記符号化規則は、前記情報系列生成手段により挿入される、前記第 1 の DC 制御ビットおよび前記第 2 の DC 制御ビットの挿入位置が、前記符号変換単位 of 1 ビット目であるか 2 ビット目であるかにかかわらず、前記第 1 の符号状態と、前記第 2 の符号状態とが同一となった時点で、前記第 1 の仮符号系列を構成する符号化ビットの合計の 2 の補数と、前記第 2 の仮符号系列を構成する符号化ビットの合計の 2 の補数とが、常に異なるように、前記情報語に前記符号語が割り当てられている

ことを特徴とする請求項 2 に記載の符号化装置。

【請求項 4】 前記第 1 の符号変換手段および前記第 2 の符号変換手段が用いる前記符号化規則は、前記情報系列生成手段により挿入される、前記第 1 の DC 制御ビットおよび前記第 2 の DC 制御ビットの挿入位置が、前記符号変換単位 of 1 ビット目である場合に、前記第 1 の符号状態と、前記第 2 の符号状態とが同一となった時点で、前記第 1 の仮符号系列を構成する符号化ビットの合計の 2 の補数と、前記第 2 の仮符号系列を構成する符号化ビットの合計の 2 の補数とが、常に異なるように、前記情報語に前記符号語が割り当てられている

ことを特徴とする請求項 2 に記載の符号化装置。

【請求項 5】 前記第 1 の符号変換手段および前記第 2 の符号変換手段が用いる前記符号化規則は、前記情報系列生成手段により挿入される、前記第 1 の DC 制御ビットおよび前記第 2 の DC 制御ビットの挿入位置が、前記符号変換単位 of 2 ビット目である場合に、前記第 1 の符号状態と、前記第 2 の符号状態とが同一となった時点で、前記第 1 の仮符号系列を構成する符号化ビットの合計の 2 の補数と、前記第 2 の仮符号系列を構成する符号化ビットの合計の 2 の補数とが、常に異なるように、前記情報語に前記符号語が割り当てられている

ことを特徴とする請求項 2 に記載の符号化装置。

【請求項 6】 前記第 1 の符号変換手段および前記第 2 の符号変換手段は、前記情報語長  $m$  を 1 ブロックとして、1 ブロック単位で符号化を実行し、

前記第 1 の符号変換手段および前記第 2 の符号変換手段が用いる前記符号化規則は、前記符号化規則を先見型の符号変換表で表現した場合、ブロック数が奇数のときには、情報系列を構成する情報ビットの合計の 2 の補数と、仮符号系列を構成する符号化ビットの合計の 2 の補数とが異なり、前記ブロック数が偶数のときには、前記情報系列を構成する情報ビットの合計の 2 の補数と、前記仮符号系列を構成する符号化ビットの合計の 2 の補数とが一致するように、前記情報語に前記符号語が割り当てられている

ことを特徴とする請求項 1 に記載の符号化装置。

【請求項 7】 前記第 1 の符号変換手段および前記第 2 の符号変換手段は、前記情報語長  $m$  を 2 ビットとし、前記符号語長  $n$  を 3 ビットとして、最小ラン制限が 1 であり、最大ラン制限が 7 である符号化規則に基づいて、前記情報語長 2 ビットを 1 ブロックとして、1 ブロック単位で符号変換を実行する

ことを特徴とする請求項 1 に記載の符号化装置。

【請求項 8】 前記第 1 の符号変換手段および前記第 2 の符号変換手段が用いる符号化規則は、前記符号化規則を有限状態型の符号変換表で表現した場合、状態数が 7 となる

ことを特徴とする請求項 7 に記載の符号化装置。

【請求項 9】 前記第 1 の符号変換手段および前記第 2 の符号変換手段は、復号時の符号語拘束長が 3 ブロックとなるように、符号変換を実行する

ことを特徴とする請求項 7 に記載の符号化装置。

【請求項 10】 前記第 1 の符号変換手段および前記第 2 の符号変換手段が用いる前記符号化規則は、最小ラン制限が 1、最大ラン制限が 7 であり、かつ、前記符号語の最小ランの最大連続数が 5 となり、前記符号化規則を符号変換表で表現した場合、その状態数が 8 となる制限を更に有する

ことを特徴とする請求項 1 の符号化装置。

【請求項 11】  $m$  ビットの情報語を  $n$  ビットの符号語に連続して変換する符号化装置の符号化方法において、

入力された情報系列に、所定の間隔で第 1 の DC 制御ビットを挿入して、第 1 の情報系列を生成するとともに、前記所定の間隔で前記第 1 の DC 制御ビットと

は異なる第2のDC制御ビットを挿入して、第2の情報系列を生成する情報系列生成ステップと、

前記情報系列生成ステップの処理により生成された前記第1の情報系列を、情報語長 $m$ 対符号語長 $n$ の変換比率で符号変換して、第1の仮符号系列を生成するとともに、前記情報系列生成ステップの処理により生成された前記第2の情報系列を、情報語長 $m$ 対符号語長 $n$ の変換比率で符号変換して、第2の仮符号系列を生成する符号変換ステップと、

前記符号変換ステップの処理により生成された前記第1の仮符号系列と前記第2の仮符号系列とのうち、いずれか一方を選択する選択ステップとを含み、

前記符号変換ステップの処理において用いられる符号化規則は、前記符号化規則を有限状態型の符号変換表で表現した場合、所定の始点状態から符号化された前記第1の仮符号系列の第1の符号状態と、前記所定の始点状態から符号化された前記第2の仮符号系列の第2の符号状態とが同一となった時点で、前記第1の仮符号系列を構成する符号化ビットの合計の2の補数と、前記第2の仮符号系列を構成する符号化ビットの合計の2の補数とが、常に異なるように、情報語に符号語が割り当てられている

ことを特徴とする符号化方法。

【請求項12】  $m$ ビットの情報語を $n$ ビットの符号語に連続して変換する処理をコンピュータに実行させるプログラムであって、

入力された情報系列に、所定の間隔で第1のDC制御ビットを挿入して、第1の情報系列を生成するとともに、前記所定の間隔で前記第1のDC制御ビットとは異なる第2のDC制御ビットを挿入して、第2の情報系列を生成する情報系列生成ステップと、

前記情報系列生成ステップの処理により生成された前記第1の情報系列を、情報語長 $m$ 対符号語長 $n$ の変換比率で符号変換して、第1の仮符号系列を生成するとともに、前記情報系列生成ステップの処理により生成された前記第2の情報系列を、情報語長 $m$ 対符号語長 $n$ の変換比率で符号変換して、第2の仮符号系列を生成する符号変換ステップと、

前記符号変換ステップの処理により生成された前記第 1 の仮符号系列と前記第 2 の仮符号系列とのうち、いずれか一方を選択する選択ステップとを含み、

前記符号変換ステップの処理において用いられる符号化規則は、前記符号化規則を有限状態型の符号変換表で表現した場合、所定の始点状態から符号化された前記第 1 の仮符号系列の第 1 の符号状態と、前記所定の始点状態から符号化された前記第 2 の仮符号系列の第 2 の符号状態とが同一となった時点で、前記第 1 の仮符号系列を構成する符号化ビットの合計の 2 の補数と、前記第 2 の仮符号系列を構成する符号化ビットの合計の 2 の補数とが、常に異なるように、情報語に符号語が割り当てられている

ことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項 1 3】  $m$  ビットの情報語を  $n$  ビットの符号語に連続して変換する処理をコンピュータに実行させるプログラムであって、

入力された情報系列に、所定の間隔で第 1 の DC 制御ビットを挿入して、第 1 の情報系列を生成するとともに、前記所定の間隔で前記第 1 の DC 制御ビットとは異なる第 2 の DC 制御ビットを挿入して、第 2 の情報系列を生成する情報系列生成ステップと、

前記情報系列生成ステップの処理により生成された前記第 1 の情報系列を、情報語長  $m$  対符号語長  $n$  の変換比率で符号変換して、第 1 の仮符号系列を生成するとともに、前記情報系列生成ステップの処理により生成された前記第 2 の情報系列を、情報語長  $m$  対符号語長  $n$  の変換比率で符号変換して、第 2 の仮符号系列を生成する符号変換ステップと、

前記符号変換ステップの処理により生成された前記第 1 の仮符号系列と前記第 2 の仮符号系列とのうち、いずれか一方を選択する選択ステップとを含み、

前記符号変換ステップの処理において用いられる符号化規則は、前記符号化規則を有限状態型の符号変換表で表現した場合、所定の始点状態から符号化された前記第 1 の仮符号系列の第 1 の符号状態と、前記所定の始点状態から符号化され



た前記第2の仮符号系列の第2の符号状態とが同一となった時点で、前記第1の仮符号系列を構成する符号化ビットの合計の2の補数と、前記第2の仮符号系列を構成する符号化ビットの合計の2の補数とが、常に異なるように、情報語に符号語が割り当てられている

ことを特徴とするプログラム。

【請求項14】 mビットの情報語をnビットの符号語に連続して変換する符号化装置において、

mビットの前記情報語をnビットの前記符号語に連続して変換する符号化手段を備え、

前記符号化手段は、最小ラン制限が1、最大ラン制限が7であり、かつ、前記符号語の最小ランの最大連続数が3乃至5となる符号化規則に基づいて、符号化処理を実行する

ことを特徴とする符号化装置。

【請求項15】 前記符号化手段が用いる前記符号化規則は、符号変換表を用いて表現した場合の状態数が8であり、前記最小ランの最大連続数が5である

ことを特徴とする請求項14に記載の符号化装置。

【請求項16】 前記符号化手段は、復号時の符号語拘束長が4ブロックとなるように、符号変換を実行する

ことを特徴とする請求項14に記載の符号化装置。

【請求項17】 mビットの情報語をnビットの符号語に連続して変換する符号化装置の符号化方法において、

最小ラン制限が1、最大ラン制限が7であり、かつ、前記符号語の最小ランの最大連続数が3乃至5となる符号化規則に基づいて、mビットの前記情報語をnビットの前記符号語に連続して変換する符号化ステップ

を含むことを特徴とする符号化方法。

【請求項18】 mビットの情報語をnビットの符号語に連続して変換する処理をコンピュータに実行させるプログラムであって、

最小ラン制限が1、最大ラン制限が7であり、かつ、前記符号語の最小ランの最大連続数が3乃至5となる符号化規則に基づいて、mビットの前記情報語をn

ビットの前記符号語に連続して変換する符号化ステップ

を含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項 19】  $m$ ビットの情報語を  $n$ ビットの符号語に連続して変換する処理をコンピュータに実行させるプログラムであって、

最小ラン制限が 1、最大ラン制限が 7 であり、かつ、前記符号語の最小ランの最大連続数が 3 乃至 5 となる符号化規則に基づいて、 $m$ ビットの前記情報語を  $n$ ビットの前記符号語に連続して変換する符号化ステップ

を含むことを特徴とするプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、符号化装置および符号化方法、記録媒体、並びにプログラムに関し、特に、符号系列の DC 制御を行うための DC 制御ビットを情報系列に挿入し、 $m$ ビットの情報語を  $n$ ビットの符号語に連続して変換する場合に用いて好適な、符号化装置および符号化方法、記録媒体、並びにプログラムに関する。

【0002】

【従来の技術】

各種記録再生装置や通信装置においては、多くの場合、入力情報を符号化することによる、デジタル伝送情報の誤り率の低減が図られている。

【0003】

図 1 に、従来の一般的な記録再生装置 1 における、デジタル信号処理回路のブロック図を示す。

【0004】

符号化部 11 は、連続的に入力される情報系列を  $m:n$  ( $m$  は符号化前のデータビット長、 $n$  は符号化後のデータビット長) の比に符号化し、2 値の記録符号系列を生成する。ここで、 $m$  は情報語長、 $n$  は符号語長、そして、 $m/n$  は符号率もしくは符号化率と称される。

【0005】

記録再生部 12 は、例えば、磁気ヘッドや光ピックアップ、およびそれらの駆動を制御するための制御回路などで構成され、符号化部 11 から入力された記録符号を、図示しない記録媒体に記録する。また、記録再生部 12 は、記録媒体に記録されている信号を再生して、アナログの再生波を、アナログ等化部 13 に出力する。アナログ等化部 13 は、記録再生部 12 から入力された再生波を所定の目標等化特性に等化する。

#### 【0006】

A/D変換部 14 は、供給された等化信号を、デジタル再生信号に変換する。A/D変換部 14 には、PLL (Phase Locked Loop: 位相同期ループ) 回路が含まれる。PLL には、位相誤差検出のみをデジタル部で行うハイブリットデジタル PLL を用いても良いし、位相誤差検出と信号同期の両方をデジタル部で行うフルデジタル PLL を用いても良い。

#### 【0007】

また、アナログ等化部 13 による等化が充分でない場合には、A/D変換部 14 と符号検出部 15 との間に、更に、デジタル等化部を設けても良い。なお、この場合は、アナログ等化部 13 に代わって、ローパスフィルタを設けるようにしても良い。

#### 【0008】

符号検出部 15 は、デジタル再生信号 (等化信号) の入力を受けて、符号に変換 (符号を検出) する。近年では、符号検出部 15 にはビタビ検出器等の軟判定検出器を用いるのが一般的である。復号部 16 は、連続的に入力される検出符号を受けて、 $n:m$  の比で検出情報に復号して出力し、検出情報系列とする。

#### 【0009】

図 1 において、符号化部 11 によって実行される符号化処理に用いられる符号として、様々な符号が実用化されている。例えば、ビット 1 で記録矩形波を反転させる NRZI (non return to zero on one) 変調を前提としたとき、符号系列における 1 と 1 との間に入る 0 の個数の最小値 (最小ラン) を  $d$ 、最大値 (最大ラン) を  $k$  に制限した、 $(d, k)$  走長制限 (RLL: run length limited) 符号は、様々な記録再生装置において広く用いられている。

## 【0010】

特に、信号再生に光ピックアップを用いる光ディスク再生装置においては、光ピックアップの非線形性による再生信号品質の劣化を低減させるために、最小ラン d を 1 以上とした最小ラン制限符号が広く用いられている。例えば、一般的に普及しているコンパクトディスク (CD: Compact Disk) の記録再生装置および音楽用のミニディスク (MD: Mini Disk) の記録再生装置では、EFM (eight to fourteen modulation) と呼ばれる符号化率  $8/17$  の (2, 10) RLL 符号が採用されており、データ記録用のミニディスク記録再生装置に用いられる MD DAT A2 においては、符号化率  $2/3$  の (1, 7) RLL 符号が採用されている。

## 【0011】

また、(d, k) RLL 符号が採用されているのは、記録再生装置のみではなく、例えば、IrDA-VFIR (Infrared Data Association, Very Fast Infrared) と呼ばれる赤外線を用いた無線通信装置においては、符号化率  $2/3$  の (1, 13) RLL 符号が採用されている。

## 【0012】

(d, k) RLL 符号のうち、例えば、EFM 符号は、固定長符号 (fixed length code) であり、(1, 7) RLL 符号は、可変長符号 (variable length code) である。可変長符号は、一般的に、先見 (look ahead) 型の符号変換表か、もしくは、有限状態 (finite state) 型の符号変換表のうちの、いずれか一方に基づいて符号化される。

## 【0013】

図 2 および図 3 は、一般的に知られている、Cohn-Jacoby-Bates による (1, 7) RLL 符号の先見型の符号変換表であり、図 2 は基本変換表、図 3 は違反時における変換表である。情報系列は、通常時においては、図 2 の基本変換表に従って符号化されるが、図 3 の違反時変換表に記載された情報系列が供給された場合、図 3 の違反時変換表が優先して用いられて、符号化処理が実行される (例えば、特許文献 1 参照)。

## 【0014】

【特許文献 1】

米国特許、M. Cohn; G. Jacoby; C. Bates, USP 4,337,458

【0 0 1 5】

1 ブロックを、符号化の最小単位（符号変換単位、例えば、図 2 の基本変換表において、入力情報 2 ビットもしくは符号 3 ビットに相当するもの）とすれば、図 2 および図 3 の符号変換表に基づいて符号化処理を行う符号化部 1 1 は、符号化処理の実行のために、1 ブロック分の情報系列を先見する必要がある。また、図 2 および図 3 に示される先見型の符号変換表は、符号化部 1 1 において必要な先見量と同じ分だけ、情報語よりも遅延した符号語を情報語に割り当てなおすことにより、有限状態型の符号変換表に変換することが可能である。

【0 0 1 6】

ここで、1 ブロック遅延した符号語を情報語に割り当てなおすとは、例えば、図 3 において、2 符号語 0 0 1. 0 0 0 は、2 情報語 1 0. 0 0 に割り当てられていることから、直前の情報語が 1 0 である状態において、符号語 0 0 1 を、情報語 0 0 に割り当てるといった操作を行うことである。

【0 0 1 7】

図 4 は、図 2 および図 3 の先見型の符号変換表から変換された、(1, 7) RLL 符号の有限状態型の 5 状態符号変換表である。図 4 において、スラッシュ ( / ) の前に示される 3 ビットは符号語、スラッシュの後に示される数字は遷移する状態を、それぞれ示している。

【0 0 1 8】

すなわち、図 4 の有限状態型の符号変換表は、図 2 および図 3 の先見型の符号変換表において、1 ブロック遅延した 3 ビット符号語を、各 2 ビット情報語に割り当てたものである（例えば、非特許文献 1）。

【0 0 1 9】

【非特許文献 1】

A. Gallopoulos, C. Heegard, and P. Siegel, "The Power Spectrum of Run-Length-Limited Codes," IEEE Trans. on Com., vol.37, no.9, pp.906-917, Sep. 1989.

【0 0 2 0】

符号化部 11 は、図 2 および図 3 の先見型の符号変換表に基づいて符号化を実行した場合、情報語の先見を行うためには、入力される情報語に割り当てられた符号語よりも 1 ブロック遅延した符号語しか出力することはできない。また、符号化部 11 は、図 4 の有限状態型の符号変換表に基づいて符号化を実行した場合、情報語の先見を行う必要はないが、図 4 の有限状態型の符号変換表は、図 2 および図 3 の符号変換表と比較して、1 ブロック遅延した符号語が情報語に割り当てられている。

#### 【0021】

すなわち、ある情報系列を、図 2 および図 3 の先見型の符号変換表に基づいて符号化した場合と、図 4 の有限状態型の符号変換表に基づいて符号化した場合とは、同一の符号系列を得ることができる。したがって、先見型の符号変換表と有限状態型の符号変換表とは、異なる符号を表現しているように見えるが、符号化規則の表記方法が異なるだけである。ただし、符号化部 11 における内部の回路構成を、符号変換表の符号化規則に忠実に従って構成することを前提とすれば、符号化部 11 は、符号化処理に、先見型と有限状態型との、いずれの符号変換表を用いるかによって、異なる回路構成を有するものとなる。

#### 【0022】

また、符号が可変長符号でなく固定長符号である場合には、情報系列の先見の必要はないので、その符号変換表は有限状態型のみとなる。有限状態型の符号変換表によって表される符号が可変長符号であるか固定長符号であるかは、符号変換表において、同一の符号語が複数の情報語に割り当てられているか、単一の情報語に割り当てられているかによって区別することができる。

#### 【0023】

また、先見型の符号変換表の設計は、発見的手法に頼らなければならないことが多いが、有限状態型の符号変換表は、ACH (Adler-Coppersmith-Hassner) アルゴリズムと呼ばれる手法を用いて、理論的に設計することができる。

#### 【0024】

ここで、有限状態型の符号変換表を設計するための ACH アルゴリズムについて説明する。ACH アルゴリズムにおいては、まず、目標とする符号語のビット数を

パラメータとして、符号の有限状態遷移図 (finite state transition diagram) に基づいて、その状態数と一致した次元を持つ近似固有ベクトルと呼ばれるベクトルを求める。次に、有限状態遷移図における各状態を近似固有ベクトルの各要素の数と同数用意した有限状態表を作成した後、状態分割 (state splitting) および状態融合 (state merging) と称されるそれぞれの操作を数回繰り返すことにより、最終的に簡略化された有限状態表を得ることができる。

#### 【 0 0 2 5 】

ACH アルゴリズムにおいて、近似固有ベクトルは多数存在し、最初に求める近似固有ベクトルの値によって、最終的に求められる有限状態表が異なる場合がある。ただし、状態分割および状態融合のそれぞれの操作過程には、複数の選択枝が存在する場合が多いため、同一の近似固有ベクトルを用いても、最終的に求められる有限状態表は必ずしも同一であるとはかぎらない。

#### 【 0 0 2 6 】

このようにして得られた有限状態表のうちの各符号語を、必要な情報語に適宜割り当てることにより、有限状態型の符号変換表を得ることができる。また、この方法に従って設計された符号は、全て、有限の符号語拘束長で、状態に依存しない (state independent) 復号が可能である。すなわち、この方法に従って設計された符号は、スライディングブロック復号が可能 (sliding-block decodable) であることが知られている。

#### 【 0 0 2 7 】

ACH アルゴリズムの詳細は、例えば、非特許文献 2 に開示されている。

#### 【 0 0 2 8 】

##### 【非特許文献 2】

R. Adler, D. Coppersmith, and M. Hassner, "Algorithms for Sliding Block Codes," IEEE Trans. on Information Theory, vol. IT-29, no. 1, pp. 5-22, Jan. 1983.

#### 【 0 0 2 9 】

図 5 は、NRZI 変調を前提とした、(1, 7) RLL を与える 8 状態の有限状態遷移図である。ただし、図 5 に示される有限状態遷移図における状態番号と有限

状態型の符号変換表の状態番号とは対応しないものとする。

#### 【0030】

図5の有限状態遷移図から求められる8次元の近似固有ベクトルとして、一般的に、 $[2, 3, 3, 3, 2, 2, 2, 1]$ 、もしくは、 $[3, 5, 5, 4, 4, 4, 3, 2]$ の2つの近似固有ベクトルが知られている。近似固有ベクトルの各要素の値は、有限状態遷移図中の各状態を最初に分割する数（いわゆる、ステートスプリッティング数）を示す。例えば、図4の5状態の符号変換表は、図2および図3の先見型の符号変換表から求めることができるとともに、近似固有ベクトルを $[3, 5, 5, 4, 4, 4, 3, 2]$ として、図5の有限状態遷移図からACHアルゴリズムにより理論的に求めることもできる。一方、近似固有ベクトルを $[2, 3, 3, 3, 2, 2, 2, 1]$ として、ACHアルゴリズムにより有限状態型の符号変換表を求めた場合、4状態の有限状態型の符号変換表を得ることができる。

#### 【0031】

図6は、Weathers-Wolfによる(1, 7) RLL符号の有限状態型の4状態符号変換表である。図6の符号変換表は、(1, 7) RLL符号における最少状態数の有限状態型の符号変換表として知られている（例えば、非特許文献3）。

#### 【0032】

##### 【非特許文献3】

A. D. Weathers and J.K. Wolf, "A New Rate 2/3 Sliding Block Code for the (1,7) Runlength Constraint with the Minimal Number of Encoder States," IEEE Trans. on Info. Theory, vol.37, no.3, pp.908-913, May 1991.

#### 【0033】

光ディスク記録再生装置または光ディスク再生装置においては、一般的に、低周波数帯域に使用されるサーボ信号の抽出のために、符号スペクトルの低周波数成分を、ある程度抑圧しておく必要がある。符号スペクトルの低周波数成分が全く抑圧されていないと、抽出されたサーボ信号に、データ領域の再生信号の低周波数成分が雑音として重畳されてしまうため、サーボ信号の品質が著しく劣化するという問題が発生する。



## 【0034】

符号スペクトルの低周波数成分を抑圧するには、符号系列の直流累積電荷（RDS: running digital sum）の振幅値（DSV: digital-sum variation）が、できるだけ小さくなるように、符号系列の直流成分制御（DC制御）を行うことが有効である。ここで、RDSとは、変調後の記録符号系列または伝送符号系列において、例えば、NRZI変調を前提とした符号系列であれば、NRZI変調後の符号系列において、 $\pm 1$ で表される符号極性が累積加算された値である。

## 【0035】

従来、符号系列のDC制御を行う具体的方法としては、主に、以下の3つの方法が用いられている。

## 【0036】

第1の方法は、まず、符号系列のDSVにかかわらずに符号化を実行してRDSを観測し、その後に、符号系列のDSVをできるだけ小さくするように、符号系列に一定間隔でDC制御ビットを挿入するという方法である。ただし、DC制御ビットが場合に依じてRLI制御ビットとなることを前提として、DC制御ビットの挿入後に、符号系列のRLI制限が守られるように、最初の符号化を行うようにしても良い。

## 【0037】

第1の方法を用いた符号には、例えば、上述した、CDやMDにおいて実用化されているEFM符号がある。ただし、EFM符号においては、ある情報系列が連続すると、(2, 10)のRLI制限を守るためにDC制御を行うことが不可能となり、DSVが有限とならない場合がある。しかしながら、一般的に使用される入力情報はランダム性が高いため、この点は実用上問題とはならない。

## 【0038】

第1の方法は、任意のRLI符号に対して適用可能であるという利点があるが、RLI制限を守るために挿入するDC制御ビットの数が多くなってしまい、DC制御ビットの影響を含めた符号の冗長度が増加しやすい。

## 【0039】

第2の方法は、符号変換表を有限状態型とし、符号系列のDSVができるだけ

小さくなるように、符号変換表を予め設計しておく方法である。第2の方法を用いた符号には、例えば、8ビットの情報語を16ビットの符号語に変換するEFMPlusと呼ばれる(2, 10) RLL符号があり、DVD (digital versatile-disk) の記録符号として実用化されている。上述したEFM符号は固定長符号であるが、EFMPlus符号は、同一の符号語が複数の情報語に割り当てられた可変長符号である。

#### 【0040】

ただし、EFMPlus符号においては、256個の情報語のうち、DC制御可能な情報語の数は95個しかなく、これら以外の161個の情報語のいずれかが連続した場合、DSVが有限とならないことがある。しかしながら、この場合においても、上述したEFM符号の場合と同様に、使用される入力情報は、一般的に、ランダム性が高いため、この点については、実用上問題とはならない。

#### 【0041】

第2の方法は、DC制御ビットの挿入が必要ないという利点があるが、一部の符号においては、高い符号化率での符号変換表の設計が難しく、また、多くの場合、符号変換表が複雑となるため、符号化器および復号器の構成が複雑になりやすい。

#### 【0042】

第3の方法は、符号変換表を先見型とし、情報語の合計の2の補数と符号語の合計の2の補数とが常に一致するように、情報語に符号語を予め割り当てておき、符号化する際に、符号系列のDSVをできるだけ小さくするように、情報系列の方に一定間隔でDC制御ビットを挿入するという方法である。第3の方法に基づいて符号語割当てを行うと、挿入するDC制御ビットが0の場合と、1の場合とで、その後に続く符号極性の反転が保証されるため、符号系列のDC制御を効率良く行うことができる。

#### 【0043】

第3の方法による符号語割当て法は、情報系列のパリティと符号系列のパリティとが常に一致するように行われることから、パリティ保存型 (PP: parity preserving) 符号語割当て (word assignment) 法と称される。

## 【0044】

すなわち、Kahlman-Imminkによって1994年に考案されたパリティ保存型符号語割当て法とは、先見型の符号変換表において、情報語の合計の2の補数をp、符号語の合計の2の補数をqとしたとき、次の式(1)が成立するように符号語を情報語に割り当てるものである(例えば、特許文献2参照)。

## 【0045】

$$p = q \quad \cdot \cdot \cdot (1)$$

## 【0046】

## 【特許文献2】

米国特許、J. Kahlman; A. Immink, USP 5,477,222

## 【0047】

第3の方法においては、情報系列にどのようなDC制御ビットを挿入しても、符号系列のRLI制限を乱すことはない。このため、第3の方法を用いると、符号スペクトルの低周波数成分について同等の抑圧効果を得ようとした場合、符号系列にDC制御ビットを挿入する第1の方法と比較して、DC制御ビットの冗長度を減少させることができることが多い。

## 【0048】

ただし、第3の方法においては、DC制御ビットの選択によって符号系列の極性反転は保証されるが、制御間隔内でのRDSの極性反転は必ずしも保証されない。このため、DC制御ビットに0もしくは1のどちらを選んでもRDSの極性が反転しないある特定の入力情報が連続すると、DSVが有限とならない場合がある。しかしながら、上述したEFM符号およびEFMPlus符号の場合と同様に、一般的に使用される入力情報はランダム性が高いため、この点は実用上大きな問題とはならない。

## 【0049】

また、情報系列にDC制御ビットを挿入するDC制御方法を用いた場合、用いられる先見型の符号変換表において、仮にパリティ保存型符号語割当てが行われていなくても、ある程度の確率で符号系列のRDSの極性を反転させることができるので、符号系列のDC制御は、全く不可能になるのではない。すなわち、パ

リティ保存型符号語割当て法を用いることの利点は、情報系列にDC制御ビットを挿入した際に符号系列のRDSの極性が反転する確率を大きく上げることにより、符号系列のDC制御をより効率良く行うことができる点である。

#### 【 0 0 5 0 】

符号変換表において、具体的にパリティ保存型符号語割当てを行う方法は、先見型の符号変換表についてしか知られていない。つまり、ACHアルゴリズムから求められる有限状態型の符号変換表については、パリティ保存型となる符号語割当て法は見い出されていない。この点については、パリティ保存型符号語割当て法の発明者の1人であるImminkが、その著書（非特許文献4）の中で明確に指摘している。

#### 【 0 0 5 1 】

##### 【非特許文献4】

K. Immink, "Codes for Mass Data Storage," Shannon Foundation publishers, Netherlands, p.290, 1999.

#### 【 0 0 5 2 】

また、上述した特許文献2に開示されている、パリティ保存型符号語割当てを行った最初の先見型の符号変換表によって表される符号は、(1, 8) RLL符号であったが、現在では(1, 7) RLL符号においても、パリティ保存型符号語割当てを行った先見型の符号変換表を設計することができることが知られている。

#### 【 0 0 5 3 】

図7および図8は、(1, 7) RLL符号の、パリティ保存型符号語割当てを行った先見型の符号変換表である。

#### 【 0 0 5 4 】

図7は、通常時における情報系列の符号化に用いられる基本変換表であり、図8は、違反時における符号化に用いられる違反時変換表である。図7の基本変換表のみに基づいて符号化を行った場合は(1, 8) RLL符号となり、それに、図8の違反時変換表を組み合わせることによって(1, 7) RLL符号とすることができる。換言すれば、(1, 7) RLL符号においては、情報系列の符号化

処理時に、違反時変換表に記載された情報系列が供給された場合、図7の基本変換表より、図8の違反時変換表が優先的に用いられる。図7の基本変換表、および、図8の違反時変換表のいずれにおいても、情報語の合計の2の補数と符号語の合計の2の補数とは全て一致している（例えば、特許文献3）。

【0055】

【特許文献3】

特開平11-346154号公報

【0056】

また、特許文献3にも記載されているように、光ディスク記録再生装置、もしくは、光ディスク再生装置において、最小ランの長い連続は、デフォーカスやタンジェンシャルチルト等が発生している状況におけるビット誤り率に悪影響を与える。このため、図7および図8の先見型の符号変換表に、違反時符号変換表を更に追加することにより、最小ランの最大連続数を6に制限する方法が提案されている。ただし、ここで、最小ランの最大連続数がRであるとは、例えば、最小ランが1の符号で、NRZI変調を前提とした場合、01のr回連続を(01)<sup>r</sup>と表現したときに、001(01)<sup>r</sup>00の符号系列において、必ず、 $r \leq R$ となっていることである。

【0057】

図9は、(1, 7) RLL符号の最小ランの最大連続数を6に制限するための、図7および図8に追加される、パリティ保存型符号語割当てを行った先見型の違反時符号変換表である。

【0058】

図9の違反時変換表においても、情報語の合計の2の補数と符号語の合計の2の補数とは一致している。図7および図8の符号変換表に図9の違反時符号変換表を加えた符号変換規則に従って符号化を行うことにより、パリティ保存型の符号語割当てを保ったまま、最小ランの最大連続数を6に制限することができる。

【0059】

図7、図8、および、図9に示された、パリティ保存型符号語割当てを行った先見型の符号変換表によって表される(1, 7) RLL符号は、次世代の光ディ

スク記録再生装置であるBlu-rayの記録符号に採用されている。

#### 【0060】

##### 【発明が解決しようとする課題】

上述したように、パリティ保存型符号語割当て法は、符号系列のDC制御を効率良く行うための優れた方法の1つであるが、具体的な符号語割当てを行う方法は、先見型の符号変換表によるものであり、かつ、その設計は、発見的手法に頼るしかない。このため、現在、パリティ保存型符号語割当てを行った具体的な符号変換表は、非常にわずかししか開示されていない。

#### 【0061】

一方、有限状態型の符号変換表は、ACHアルゴリズムを用いて、理論的に設計することができる。したがって、有限状態型の符号変換表において、先見型の符号変換表におけるパリティ保存型符号語割当て法と同様の効果を有する符号語割り当て法を見出すことができれば、DC制御効率に優れた符号の符号変換表を理論的に設計することができる。

#### 【0062】

また、従来、可変長符号において、先見型の符号変換表を有限状態型の符号変換表に変換する技術については開示されているが、有限状態型の符号変換表を先見型の符号変換表に変換する技術については開示されておらず、その具体的方法は知られていない。有限状態型の符号変換表を、先見型の符号変換表に変換することができれば、上述した有限状態型の符号変換表における符号語割当て法と、従来のパリティ保存型符号語割当て法との差異を、容易に明確化することができる。

#### 【0063】

また、符号化部11が、これまでに開示されているパリティ保存型符号語割当てを行った先見型の符号変換表の符号化規則に従って符号化を行った場合、従来の符号化方法を用いた場合と比較して、その構成が複雑になってしまう。

#### 【0064】

例えば、図7および図8の先見型の符号変換表においては、必要な情報系列の先見量が3ブロックであることから、3ブロック遅延した符号語を、情報語に割

り当てなおすことにより有限状態型の符号変換表に変換すると、その状態数は104となる。符号化部11においては、必ずしも有限状態型の符号変換表の符号化規則に従う必要はないが、有限状態型の符号変換表に変換したときに状態数が多いということは、たとえ先見型の符号変換表であっても、符号化規則がより複雑ということであるため、符号化部11の構成は、一般的に複雑となる。

#### 【0065】

このことは、最小ランの最大連続数を有限に制限した場合においては、特に顕著である。例えば、図7および図8を用いて説明した先見型の符号変換表に、更に、図9の先見型の符号変換表を組み合わせた場合、必要な情報系列の先見量が5ブロックとなることから、5ブロック遅延した符号語を情報語に割当てなおすことにより、有限状態型の符号変換表に変換すると、その状態数は、1691となり、状態数が非常に多くなってしまう。

#### 【0066】

更に、これまでに開示されているパリティ保存型符号語割当てを行った先見型の符号変換表に基づいて符号化を行った場合、復号時の符号語拘束長が、例えば、図2および図3の符号変換表、並びに、図6の符号変換表に従って符号化を行ったときの復号時の符号語拘束長よりも長くなってしまいう問題が発生する。ここで、復号時の符号語拘束長とは、スライディングブロック復号のために保持が必要な符号語のブロック数である。

#### 【0067】

復号時の符号語拘束長が長くなると、復号部16の構成が複雑となるだけでなく復号時の誤り伝播長が長くなり、復号後のビット誤り率が劣化してしまう。ここで、復号時の誤り伝播長とは、符号系列における1ビットの誤りが情報系列に伝播する可能性のある最大のビット数である。最小ランの最大連続数を有限に制限した場合においても、この問題は同様に発生する。

#### 【0068】

図10に、図2乃至図9を用いて説明した(1, 7)RL符号を用いた場合における、符号化時の状態数、復号時の符号語拘束長、および、復号時の誤り伝播長の計算結果を示す。ただし、図10において、図7および図8を組み合わせ

た符号変換表、並びに、図7、図8および図9を組み合わせた符号変換表の、それぞれの状態数は、これら先見型の符号変換表を、上述した方法によって有限状態型の符号変換表に変換したときに生じる状態数である。

#### 【0069】

図10に示されるように、パリティ保存型符号語割当てを行った図7および図8、もしくは、図7、図8および図9を組み合わせた符号変換表の符号化規則に従って符号化される(1, 7) RLL符号は、その符号化時の状態数が104、もしくは1691となり、図2および図3の符号変換表、または図4の符号変換表を用いて説明した場合の状態数5や、図6の符号変換表を用いて説明した場合の状態数4と比較して、20倍以上となる。また、パリティ保存型符号語割当てを行った図7および図8、もしくは、図7、図8および図9を組み合わせた符号変換表の符号化規則に従って符号化される(1, 7) RLL符号は、復号時の符号語拘束長および復号時の誤り伝播長においても、図2および図3の符号変換表、または図4の符号変換表や、図6の符号変換表を用いた場合の2倍程度となってしまう、そのいずれもが大きく劣化していることが分かる。

#### 【0070】

すなわち、これまでに知られているパリティ保存型符号語割当てを行った先見型の符号変換表によって表される符号は、その符号化時の状態数、復号時の符号語拘束長、および復号時の誤り伝播長の、いずれの値も、パリティ保存型符号語割当てを行っていない符号よりも、大きく劣化しているため、これらの値を改善することが望まれていた。

#### 【0071】

また、最小ランの最大連続数をより短くすることができれば、光ディスク(記録)再生装置におけるデフォーカスやタンジェンシャルチルト等の状況におけるビット誤り率を、より効果的に改善することができる。しかしながら、(1, 7) RLL符号において、これまでに達成されている最小ランの最大連続数の最小値は6であった。(1, 7) RLL符号における最小ランの最大連続数を、6よりも短くするための技術は、現在、開示されていない。

#### 【0072】



本発明はこのような状況に鑑みてなされたものであり、符号系列のDC制御を行うためのDC制御ビットを情報系列に挿入し、mビットの情報語をnビットの符号語に連続して変換する符号化方法において、符号化時の状態数を従来よりも少なくし、かつ、復号時の符号語拘束長、および復号時の誤り伝播長を、従来における場合よりも短くすることができるようにし、(1, 7) RLL符号における最小ランの最大連続数を、従来の6よりも短くすることができるようにするものである。

#### 【0073】

##### 【課題を解決するための手段】

本発明の第1の符号化装置は、入力された情報系列に、所定の間隔で第1のDC制御ビットを挿入して、第1の情報系列を生成するとともに、所定の間隔で第1のDC制御ビットとは異なる第2のDC制御ビットを挿入して、第2の情報系列を生成する情報系列生成手段と、情報系列生成手段により生成された第1の情報系列を、情報語長m対符号語長nの変換比率で符号変換して、第1の仮符号系列を生成する第1の符号変換手段と、情報系列生成手段により生成された第2の情報系列を、情報語長m対符号語長nの変換比率で符号変換して、第2の仮符号系列を生成する第2の符号変換手段と、第1の符号変換手段により生成された第1の仮符号系列と第2の符号変換手段により生成された第2の仮符号系列とのうち、いずれか一方を選択する選択手段とを備え、第1の符号変換手段および第2の符号変換手段が用いる符号化規則は、符号化規則を有限状態型の符号変換表で表現した場合、所定の始点状態から符号化された第1の仮符号系列の第1の符号状態と、所定の始点状態から符号化された第2の仮符号系列の第2の符号状態とが同一となった時点で、第1の仮符号系列を構成する符号化ビットの合計の2の補数と、第2の仮符号系列を構成する符号化ビットの合計の2の補数とが、常に異なるように、情報語に符号語が割り当てられていることを特徴とする。

#### 【0074】

第1の符号変換手段および第2の符号変換手段には、情報語長mを2ビットとして、2ビットを符号変換単位として符号変換を実行させるようにすることができる。

## 【 0 0 7 5 】

第 1 の符号変換手段および第 2 の符号変換手段が用いる符号化規則は、情報系列生成手段により挿入される、第 1 の DC 制御ビットおよび第 2 の DC 制御ビットの挿入位置が、符号変換単位の 1 ビット目であるか 2 ビット目であるかにかかわらず、第 1 の符号状態と、第 2 の符号状態とが同一となった時点で、第 1 の仮符号系列を構成する符号化ビットの合計の 2 の補数と、第 2 の仮符号系列を構成する符号化ビットの合計の 2 の補数とが、常に異なるように、情報語に符号語が割り当てられているものとすることができる。

## 【 0 0 7 6 】

第 1 の符号変換手段および第 2 の符号変換手段が用いる符号化規則は、情報系列生成手段により挿入される、第 1 の DC 制御ビットおよび第 2 の DC 制御ビットの挿入位置が、符号変換単位の 1 ビット目である場合に、第 1 の符号状態と、第 2 の符号状態とが同一となった時点で、第 1 の仮符号系列を構成する符号化ビットの合計の 2 の補数と、第 2 の仮符号系列を構成する符号化ビットの合計の 2 の補数とが、常に異なるように、情報語に符号語が割り当てられているものとすることができる。

## 【 0 0 7 7 】

第 1 の符号変換手段および第 2 の符号変換手段が用いる符号化規則は、情報系列生成手段により挿入される、第 1 の DC 制御ビットおよび第 2 の DC 制御ビットの挿入位置が、符号変換単位の 2 ビット目である場合に、第 1 の符号状態と、第 2 の符号状態とが同一となった時点で、第 1 の仮符号系列を構成する符号化ビットの合計の 2 の補数と、第 2 の仮符号系列を構成する符号化ビットの合計の 2 の補数とが、常に異なるように、情報語に符号語が割り当てられているものとすることができる。

## 【 0 0 7 8 】

第 1 の符号変換手段および第 2 の符号変換手段には、情報語長  $m$  を 1 ブロックとして、1 ブロック単位で符号化を実行させるようにすることができ、第 1 の符号変換手段および第 2 の符号変換手段が用いる符号化規則は、符号化規則を先見型の符号変換表で表現した場合、ブロック数が奇数のときには、情報系列を構成

する情報ビットの合計の2の補数と、仮符号系列を構成する符号化ビットの合計の2の補数とが異なり、ブロック数が偶数のときには、情報系列を構成する情報ビットの合計の2の補数と、仮符号系列を構成する符号化ビットの合計の2の補数とが一致するように、情報語に符号語が割り当てられているものとすることができる。

#### 【0079】

第1の符号変換手段および第2の符号変換手段には、情報語長 $m$ を2ビットとし、符号語長 $n$ を3ビットとして、最小ラン制限が1であり、最大ラン制限が7である符号化規則に基づいて、情報語長2ビットを1ブロックとして、1ブロック単位で符号変換を実行させるようにすることができる。

#### 【0080】

第1の符号変換手段および第2の符号変換手段が用いる符号化規則は、符号化規則を有限状態型の符号変換表で表現した場合、状態数が7となるものとすることができる。

#### 【0081】

第1の符号変換手段および第2の符号変換手段には、復号時の符号語拘束長が3ブロックとなるように、符号変換を実行させるようにすることができる。

#### 【0082】

第1の符号変換手段および第2の符号変換手段が用いる符号化規則は、最小ラン制限が1、最大ラン制限が7であり、かつ、符号語の最小ランの最大連続数が5となり、符号化規則を符号変換表で表現した場合、その状態数が8となる制限を更に有するものとすることができる。

#### 【0083】

本発明の第1の符号化方法は、入力された情報系列に、所定の間隔で第1のDC制御ビットを挿入して、第1の情報系列を生成するとともに、所定の間隔で第1のDC制御ビットとは異なる第2のDC制御ビットを挿入して、第2の情報系列を生成する情報系列生成ステップと、情報系列生成ステップの処理により生成された第1の情報系列を、情報語長 $m$ 対符号語長 $n$ の変換比率で符号変換して、第1の仮符号系列を生成するとともに、情報系列生成ステップの処理により生成

された第 2 の情報系列を、情報語長  $m$  対符号語長  $n$  の変換比率で符号変換して、第 2 の仮符号系列を生成する符号変換ステップと、符号変換ステップの処理により生成された第 1 の仮符号系列と第 2 の仮符号系列とのうち、いずれか一方を選択する選択ステップとを含み、符号変換ステップの処理において用いられる符号化規則は、符号化規則を有限状態型の符号変換表で表現した場合、所定の始点状態から符号化された第 1 の仮符号系列の第 1 の符号状態と、所定の始点状態から符号化された第 2 の仮符号系列の第 2 の符号状態とが同一となった時点で、第 1 の仮符号系列を構成する符号化ビットの合計の 2 の補数と、第 2 の仮符号系列を構成する符号化ビットの合計の 2 の補数とが、常に異なるように、情報語に符号語が割り当てられていることを特徴とする。

#### 【 0 0 8 4 】

本発明の第 1 の記録媒体に記録されているプログラムは、入力された情報系列に、所定の間隔で第 1 の DC 制御ビットを挿入して、第 1 の情報系列を生成するとともに、所定の間隔で第 1 の DC 制御ビットとは異なる第 2 の DC 制御ビットを挿入して、第 2 の情報系列を生成する情報系列生成ステップと、情報系列生成ステップの処理により生成された第 1 の情報系列を、情報語長  $m$  対符号語長  $n$  の変換比率で符号変換して、第 1 の仮符号系列を生成するとともに、情報系列生成ステップの処理により生成された第 2 の情報系列を、情報語長  $m$  対符号語長  $n$  の変換比率で符号変換して、第 2 の仮符号系列を生成する符号変換ステップと、符号変換ステップの処理により生成された第 1 の仮符号系列と第 2 の仮符号系列とのうち、いずれか一方を選択する選択ステップとを含み、符号変換ステップの処理において用いられる符号化規則は、符号化規則を有限状態型の符号変換表で表現した場合、所定の始点状態から符号化された第 1 の仮符号系列の第 1 の符号状態と、所定の始点状態から符号化された第 2 の仮符号系列の第 2 の符号状態とが同一となった時点で、第 1 の仮符号系列を構成する符号化ビットの合計の 2 の補数と、第 2 の仮符号系列を構成する符号化ビットの合計の 2 の補数とが、常に異なるように、情報語に符号語が割り当てられていることを特徴とする。

#### 【 0 0 8 5 】

本発明の第 1 のプログラムは、入力された情報系列に、所定の間隔で第 1 の D

C 制御ビットを挿入して、第 1 の情報系列を生成するとともに、所定の間隔で第 1 の D C 制御ビットとは異なる第 2 の D C 制御ビットを挿入して、第 2 の情報系列を生成する情報系列生成ステップと、情報系列生成ステップの処理により生成された第 1 の情報系列を、情報語長  $m$  対符号語長  $n$  の変換比率で符号変換して、第 1 の仮符号系列を生成するとともに、情報系列生成ステップの処理により生成された第 2 の情報系列を、情報語長  $m$  対符号語長  $n$  の変換比率で符号変換して、第 2 の仮符号系列を生成する符号変換ステップと、符号変換ステップの処理により生成された第 1 の仮符号系列と第 2 の仮符号系列とのうち、いずれか一方を選択する選択ステップとを含み、符号変換ステップの処理において用いられる符号化規則は、符号化規則を有限状態型の符号変換表で表現した場合、所定の始点状態から符号化された第 1 の仮符号系列の第 1 の符号状態と、所定の始点状態から符号化された第 2 の仮符号系列の第 2 の符号状態とが同一となった時点で、第 1 の仮符号系列を構成する符号化ビットの合計の 2 の補数と、第 2 の仮符号系列を構成する符号化ビットの合計の 2 の補数とが、常に異なるように、情報語に符号語が割り当てられていることを特徴とする。

#### 【 0 0 8 6 】

本発明の第 1 の符号化装置および符号化方法、並びにプログラムにおいては、入力された情報系列に、所定の間隔で第 1 の D C 制御ビットが挿入されて、第 1 の情報系列が生成され、所定の間隔で第 1 の D C 制御ビットとは異なる第 2 の D C 制御ビットが挿入されて、第 2 の情報系列が生成され、生成された第 1 の情報系列が、情報語長  $m$  対符号語長  $n$  の変換比率で符号変換されて、第 1 の仮符号系列が生成され、生成された第 2 の情報系列が、情報語長  $m$  対符号語長  $n$  の変換比率で符号変換されて、第 2 の仮符号系列が生成され、生成された第 1 の仮符号系列と第 2 の仮符号系列とのうち、いずれか一方が選択され、符号変換において用いられる符号化規則は、符号化規則を有限状態型の符号変換表で表現した場合、所定の始点状態から符号化された第 1 の仮符号系列の第 1 の符号状態と、所定の始点状態から符号化された第 2 の仮符号系列の第 2 の符号状態とが同一となった時点で、第 1 の仮符号系列を構成する符号化ビットの合計の 2 の補数と、第 2 の仮符号系列を構成する符号化ビットの合計の 2 の補数とが、常に異なるように、

情報語に符号語が割り当てられている。

【0087】

本発明の第2の符号化装置は、mビットの情報語をnビットの符号語に連続して変換する符号化手段を備え、符号化手段は、最小ラン制限が1、最大ラン制限が7であり、かつ、符号語の最小ランの最大連続数が3乃至5となる符号化規則に基づいて、符号化処理を実行することを特徴とする。

【0088】

符号化手段が用いる符号化規則は、符号変換表を用いて表現した場合の状態数が8であり、最小ランの最大連続数が5であるものとすることができる。

【0089】

符号化手段には、復号時の符号語拘束長が4ブロックとなるように、符号変換を実行させるようにすることができる。

【0090】

本発明の第2の符号化方法は、最小ラン制限が1、最大ラン制限が7であり、かつ、符号語の最小ランの最大連続数が3乃至5となる符号化規則に基づいて、mビットの情報語をnビットの符号語に連続して変換する符号化ステップを含むことを特徴とする。

【0091】

本発明の第2の記録媒体に記録されているプログラムは、最小ラン制限が1、最大ラン制限が7であり、かつ、符号語の最小ランの最大連続数が3乃至5となる符号化規則に基づいて、mビットの情報語をnビットの符号語に連続して変換する符号化ステップを含むことを特徴とする。

【0092】

本発明の第2のプログラムは、最小ラン制限が1、最大ラン制限が7であり、かつ、符号語の最小ランの最大連続数が3乃至5となる符号化規則に基づいて、mビットの情報語をnビットの符号語に連続して変換する符号化ステップを含むことを特徴とする。

【0093】

本発明の第2の符号化装置および符号化方法、並びにプログラムにおいては、

最小ラン制限が1、最大ラン制限が7であり、かつ、符号語の最小ランの最大連続数が3乃至5となる符号化規則に基づいて、mビットの情報語がnビットの符号語に連続して変換される。

【0094】

【発明の実施の形態】

以下、図を参照して、本発明の実施の形態について説明する。

【0095】

図11は、本発明を適応した記録再生装置31の構成を示すブロック図である。なお、従来の場合と対応する部分には同一の符号を付してあり、その説明は適宜省略する。すなわち、図11の記録再生装置31は、符号化部11に代わって、符号化部51が設けられている以外は、図1における場合と同様の構成を有している。

【0096】

図11に示される記録再生装置31は、符号化部51、および記録再生部12から構成される符号化装置41と、記録再生部12、アナログ等化回路4、A/D変換部14、符号検出部15、および復号部16から構成される復号装置42との2つの装置として構成するようにしても良い。なお、復号装置42は、符号化装置41により符号化され、図示しないネットワークに伝送された伝送符号列を取得して、復号するようにしても良い。

【0097】

また、図11に示される記録再生装置31、または、復号装置42においては、図1を用いて説明した従来における場合と同様に、アナログ等化部13による等化が充分でない場合、A/D変換部14と符号検出部15との間に、更に、デジタル等化部を設けるようにしても良い。なお、この場合は、アナログ等化部13に代わってローパスフィルタを設けるようにしても良い。

【0098】

次に、図11の符号化部51が実行する、情報系列にDC制御ビットを挿入して符号化することにより、符号系列のDC制御を行う処理について説明する。

【0099】

図12は、情報系列にDC制御ビットを挿入することによって符号系列のDC制御を行う、図11の符号化部51の構成を示すブロック図である。

【0100】

DC制御ビット挿入部101は、バイト（8ビット）単位で、連続的に情報系列の入力を受ける。DC制御ビット挿入部101は、予め設定されたDC制御間隔 $T_{dc}$ に従い、仮のDC制御ビット0が挿入された情報系列 $d_0$ 、および、仮の制御ビット1が挿入された情報系列 $d_1$ を生成し、情報系列 $d_0$ を $m/n$ 変換部102に、情報系列 $d_1$ を $m/n$ 変換部103に、それぞれ供給する。

【0101】

DC制御ビット挿入部101に供給された情報系列に、 $T_{dc}$ ビットの間隔でDC制御ビットが挿入された場合、その出力系列は入力系列に対して $1/(T_{dc}+1)$ のDC制御冗長度（DC-control redundancy）を有する。

【0102】

DC制御ビット挿入部101は、更に、DC制御時刻を示すDC制御信号を、 $m/n$ 変換部102、 $m/n$ 変換部103、および符号選択信号生成部106に出力する。

【0103】

$m/n$ 変換部102は、供給された情報系列 $d_0$ を、 $m:n$ の変換比で、仮符号系列 $c_0$ に変換し、仮符号系列 $c_0$ を、 $n$ ビット単位で、遅延部104および符号選択信号生成部106に出力する。同様に、 $m/n$ 変換部103は、供給された情報系列 $d_1$ を、 $m:n$ の変換比で、仮符号系列 $c_1$ に変換し、仮符号系列 $c_1$ を、 $n$ ビット単位で、遅延部105および符号選択信号生成部106に出力する。

【0104】

遅延部104は、供給された仮符号系列 $c_0$ を、DC制御間隔分だけ遅延して符号選択部107に出力する。同様に、遅延部105は、供給された仮符号系列 $c_1$ を、DC制御間隔分だけ遅延して符号選択部107に出力する。

【0105】

符号選択信号生成部106は、供給された仮符号系列 $c_0$ および仮符号系列 $c_1$



のRDSを計算した後、符号系列のDSVをできるだけ小さくするために選択すべき仮符号は、仮符号系列 $c_0$ または仮符号系列 $c_1$ のうちのいずれの仮符号系列であるかを示す符号選択信号を生成し、入力されるDC制御信号のタイミングに従って、 $m/n$ 変換部102および $m/n$ 変換部103、並びに、符号選択部107に出力する。

#### 【0106】

符号選択部107は、遅延された仮符号系列 $c_0$ および仮符号系列 $c_1$ の供給を受け、符号選択信号生成部106から入力される符号選択信号を基に、符号系列のDSVをできるだけ小さくすることができる仮符号系列を、いずれか一方だけ選択して、符号系列として出力する。

#### 【0107】

ここで、 $m/n$ 変換部102および $m/n$ 変換部103においては、毎時刻の符号状態を保存するための内部レジスタを内部にそれぞれ有し、入力されるDC制御信号および符号選択信号を基に、仮符号系列 $c_0$ が選択された場合、 $m/n$ 変換部102の符号状態が $m/n$ 変換部103の内部レジスタに、仮符号系列 $c_1$ が選択された場合には $m/n$ 変換部103の符号状態が $m/n$ 変換部102の内部レジスタに、それぞれコピーされるようになされている。

#### 【0108】

次に、図13のフローチャートを参照して、符号化部51が実行する符号化処理について説明する。ここでは、符号化部51は、バイト数 $N_i$ の情報系列の入力を受け、ビット数 $T_{dc}$ の間隔で、DC制御ビットを挿入することにより、符号系列のDC制御を行うことができるようになされているものとする。

#### 【0109】

ステップS1において、DC制御ビット挿入部101は、変数 $i$ および変数 $j$ を0に初期化し、 $m/n$ 変換部102は、内部に有するレジスタの状態 $s_0$ に、初期状態 $s_i$ を代入し、同様に、 $m/n$ 変換部103は、内部に有するレジスタの状態 $s_1$ に、初期状態 $s_i$ を代入する。ここで、変数 $i$ は、処理済のバイト数を示す変数であり、変数 $j$ は、DC制御ビットを正しい挿入間隔で挿入するために、ビット数をカウントする変数である。

## 【0110】

DC制御ビット挿入部101は、ステップS2において、1バイトの情報系列を取得し、ステップS3において、変数*i*を、 $i = i + 1$ 、変数*j*を、 $j = j + 8$ とする。

## 【0111】

ステップS4において、DC制御ビット挿入部101は、変数*j*は、 $j < T_{dc}$ を満たすか否かを判断する。

## 【0112】

ステップS4において、変数*j*は、 $j < T_{dc}$ を満たさないと判断された場合、ステップS5において、DC制御ビット挿入部101は、供給された情報系列に、DC制御ビット0を挿入して情報系列*d*<sub>0</sub>を生成し、*m*/*n*変換部102に出力するとともに、供給された情報系列に、DC制御ビット1を挿入して情報系列*d*<sub>1</sub>を生成し、*m*/*n*変換部103に出力し、更に、変数*j*を、 $j = j - T_{dc}$ とする。

## 【0113】

ステップS4において、変数*j*は、 $j < T_{dc}$ を満たすと判断された場合、または、ステップS5の処理の終了後、ステップS6において、*m*/*n*変換部102および*m*/*n*変換部103は、供給された、情報系列*d*<sub>0</sub>または情報系列*d*<sub>1</sub>の2系列で、符号化率*m*/*n*の符号化を、それぞれ実行する。*m*/*n*変換部102は、生成した仮符号系列*c*<sub>0</sub>を、符号選択信号生成部106および遅延部104に出力し、*m*/*n*変換部103は、生成した仮符号系列*c*<sub>1</sub>を、符号選択信号生成部106および遅延部105に出力する。

## 【0114】

ステップS7において、符号選択信号生成部106は、供給された仮符号系列*c*<sub>0</sub>および仮符号系列*c*<sub>1</sub>の2系列において、それぞれDSVを計算し、仮符号系列*c*<sub>0</sub>のDSVをDSV<sub>0</sub>とし、仮符号系列*c*<sub>1</sub>のDSVをDSV<sub>1</sub>とする。

## 【0115】

ステップS8において、符号選択信号生成部106は、ステップS7の算出結果を比較して、DSV<sub>0</sub> > DSV<sub>1</sub>であるか否かを判断する。

## 【0116】

ステップS8において、 $DSV_0 > DSV_1$ ではないと判断された場合、符号選択信号生成部106は、仮符号 $c_0$ を出力符号系列として選択することを示す符号選択信号を、 $m/n$ 変換部102、 $m/n$ 変換部103、および符号選択部107に供給するので、ステップS9において、符号選択部107は、出力符号系列 $c_2$ を $c_0$ とし、 $m/n$ 変換部103は、内部のレジスタの状態 $s_1$ を、 $m/n$ 変換部102の内部のレジスタの状態 $s_0$ で置き換える。

## 【0117】

ステップS8において、 $DSV_0 > DSV_1$ であると判断された場合、符号選択信号生成部106は、仮符号 $c_1$ を出力符号系列として選択することを示す符号選択信号を、 $m/n$ 変換部102、 $m/n$ 変換部103、および符号選択部107に供給するので、ステップS10において、符号選択部107は、出力符号系列 $c_2$ を $c_1$ とし、 $m/n$ 変換部102は、内部のレジスタの状態 $s_0$ を、 $m/n$ 変換部103の内部のレジスタの状態 $s_1$ で置き換える。

## 【0118】

ステップS11において、符号選択部107は、符号系列 $c_2$ を出力する。

## 【0119】

ステップS12において、DC制御ビット挿入部101は、変数 $i$ と、供給されるバイト数 $N_i$ とを比較し、変数 $i$ が、 $i < N_i$ を満たしているか否かを判断する。ステップS12において、変数 $i$ が、 $i < N_i$ を満たしていると判断された場合、処理は、ステップS2に戻り、それ以降の処理が繰り返される。ステップS12において、変数 $i$ が、 $i < N_i$ を満たしていないと判断された場合、処理が終了される。

## 【0120】

このような処理により、 $T_{dc}$ ビットのビット間隔で、仮のDC制御ビット0が挿入され、符号化された仮符号系列 $c_0$ と、仮のDC制御ビット1が挿入され、符号化された仮符号系列 $c_1$ とが生成され、それらの $DSV$ の計算結果により、いずれかの仮符号系列が選択されて、符号系列 $c_2$ として出力されるので、符号系列のDC制御を行うことができる。

## 【0121】

そして、図13のステップS6において実行される符号化処理において用いられる符号変換表については、上述したように、パリティ保存型符号語割当てを行う方法は、これまで先見型の符号変換表についてしか知られておらず、その設計は発見的手法に頼るしかない。しかしながら、有限状態型の符号変換表において、パリティ保存型符号語割当て法と同様の効果を有する符号語割当て法を得ることができれば、ACHアルゴリズムを利用した理論的な符号変換表の設計が可能となるため、従来の符号よりも優れた符号を得ることができる。

## 【0122】

次に、図12を用いて説明した符号化部51において実行されるように、情報系列に、所定のビット間隔でDC制御ビットを挿入し、有限状態型の符号変換表の符号化規則に従って情報系列の符号化を行う場合に、符号系列のDC制御を効率的に行うことができる符号変換表に求められる特性について説明する。

## 【0123】

図12および図13を用いて説明したように、例えば、情報系列の先頭などの特定位置に、仮のDC制御ビット0を挿入した情報系列を情報系列 $d_0$ 、仮のDC制御ビット1を挿入した情報系列を情報系列 $d_1$ とした場合、これら2つの情報系列 $d_0$ および情報系列 $d_1$ を、同一状態を始点として有限状態型の符号変換表の符号化規則に従って符号化した仮の符号系列は、それぞれ仮符号系列 $c_0$ および仮符号系列 $c_1$ であり、その場合、 $m/n$ 変換部102および $m/n$ 変換部103の内部のレジスタに保存される、符号化時の符号状態は、状態 $s_0$ および状態 $s_1$ である。

## 【0124】

例えば、図6を用いて説明した符号変換表を用いて符号化が実行された場合、情報系列として[1, 1, 0, 0, 0, 1, 0]の7ビットが入力されたとき、その先頭に仮のDC制御ビットを挿入した情報系列 $d_0$ および情報系列 $d_1$ 、状態2を始点として図6の符号変換表を用いて符号化した仮符号系列 $c_0$ および仮符号系列 $c_1$ 、並びに、符号化時の符号状態 $s_0$ および符号状態 $s_1$ は、図14に示されるように、情報系列 $d_0$ : 01, 10, 00, 10、符号系列 $c_0$ : 100,

1 0 1, 0 0 0, 0 1 0、符号状態  $s_0$ : 2, 3, 1, 3、情報系列  $d_1$ : 1 1, 1 0, 0 0, 1 0、符号系列  $c_1$ : 1 0 1, 0 1 0, 0 0 0, 0 1 0、符号状態  $s_1$ : 4, 3, 1, 3 となる。

#### 【0 1 2 5】

図 1 4 に示される符号化において、各仮符号系列の状態は、最初に入力された情報語が 1 ビット異なるために、初期状態が同一であっても、状態  $s_0$  と状態  $s_1$  とは異なる状態となり、 $s_0=2$  と  $s_1=4$  となる。しかし、その後、2 ブロック目で各符号系列の状態が  $s_0=s_1=3$  で同一となっており、状態が、一旦同一になると、その後続く 3 ブロック目以降のそれぞれの仮符号系列と状態とは、同一となる。

#### 【0 1 2 6】

このように、任意の情報系列に、仮の DC 制御ビット 0 を挿入した情報系列  $d_0$ 、および、仮の DC 制御ビット 1 を挿入した情報系列  $d_1$  を、それぞれ符号化した仮符号系列  $c_0$  および仮符号系列  $c_1$  と、その状態  $s_0$  および状態  $s_1$  は、初期状態を同一としても、最初に入力される情報語（すなわち、仮の DC 制御ビット）が 1 ビット異なるために、多くの場合、一旦異なる状態となる。しかしながら、その後の変換において、状態  $s_0$  および状態  $s_1$  が同一状態になると、その後続く情報系列が何であろうと、対応する仮符号系列と状態は、2 系統のそれぞれで同一となる。

#### 【0 1 2 7】

したがって、状態  $s_0$  および状態  $s_1$  が同一となった時点において、仮符号系列  $c_0$  および仮符号系列  $c_1$  の NRZI 変調後の極性が反転していれば、その後続く仮符号系列  $c_0$  および仮符号系列  $c_1$  の RDS の正負は必ず反転するため、符号系列の DSV をできるだけ小さくするように、仮符号系列  $c_0$  または仮符号系列  $c_1$  のいずれか一方を、最終的な符号系列として選択すれば、符号系列の DC 制御を効率的に行うことができる。

#### 【0 1 2 8】

ここで、仮符号系列  $c_0$  および仮符号系列  $c_1$  の状態  $s_0$  および状態  $s_1$  が同一となった時点において、仮符号系列  $c_0$  および仮符号系列  $c_1$  の NRZI 変調後の極性を

反転させるためには、仮符号系列の状態が同一となった時点での仮符号系列  $c_0$  の合計の 2 の補数であるパリティ  $q_0$  の値と、仮符号系列  $c_1$  の合計の 2 の補数であるパリティ  $q_1$  の値とが異なっていれば良い。

【0 1 2 9】

すなわち、有限状態型の符号変換表に基づいて情報系列の符号化を行う場合、情報系列に一定間隔で DC 制御ビットを挿入することによって符号系列の DC 制御を効率良く行うためには、第 1 の挿入ビットが挿入された情報系列を、任意の始点状態から仮符号化した仮符号系列  $c_0$  と、第 2 の挿入ビットが挿入された情報系列を、同一の始点状態から仮符号化した仮符号系列  $c_1$  において、状態  $s_0$  および状態  $s_1$  が同一となった時点での、パリティ  $q_0$  およびパリティ  $q_1$  が、次の式 (2) を常に満たすようにすれば良い。

【0 1 3 0】

$$q_0 \neq q_1 \quad \dots (2)$$

【0 1 3 1】

ただし、状態  $s_0$  および状態  $s_1$  が同一となった後の、仮符号系列  $c_0$  および仮符号系列  $c_1$  は同一であるから、状態  $s_0$  および状態  $s_1$  が同一となった時点で式 (2) が成立していれば、状態  $s_0$  および状態  $s_1$  が同一となった後の仮符号系列を含めた仮符号系列のパリティ  $q_0$  およびパリティ  $q_1$  は、式 (2) を満たす。

【0 1 3 2】

上述した、図 1 4 における場合においては、状態  $s_0$  および状態  $s_1$  が同一となった 2 ブロック目までの符号系列  $c_0$  である [1, 0, 0, 1, 0, 1] の合計の 2 の補数  $q_0$  は 1 であり、同様に、2 ブロック目までの符号系列  $c_1$  である [1, 0, 1, 0, 1, 0] の合計の 2 の補数  $q_1$  は 1 であるから、この場合、式 (2) は成立していない。すなわち、DC 制御ビットの挿入位置を情報語の第 1 ビット目とした場合、図 6 の符号変換表において、任意の情報系列について式 (2) が成り立つわけではないことが分かる。

【0 1 3 3】

また、図 6 の符号変換表において、DC 制御ビットの挿入位置を、情報語の第 2 ビット目とした場合においても、同様にして、ある情報系列について式 (2)

が成り立たない。

#### 【0134】

ここでは、単一の情報系列について、式(2)が成り立たない場合について説明したが、ある情報系列について式(2)が成り立っていた場合には、その他の任意の情報系列についても式(2)が成り立つか否かを調べる必要がある。

#### 【0135】

すなわち、ある特定位置のビット（たとえば先頭ビット）を0とするブロック長 $L$ の情報系列 $d_0$ と、特定位置のビットを1とする情報系列 $d_1$ とにおいて、これら2つの情報系列を、始点状態を固定して符号化することにより得られる仮符号系列 $c_0$ および仮符号系列 $c_1$ で、それぞれの仮符号系列の状態が、始点を除いて最初に同一となるまでの、それぞれの仮符号系列の合計の2の補数 $q_0$ および $q_1$ を求め、それらが異なっているか否かを、発生し得る $L$ ブロックの全ての情報系列と全ての始点状態について調べる。

#### 【0136】

ただし、 $L$ よりも短いブロック数で状態が同一となった場合、状態が同一となるまでの情報系列が同一で、それよりも長い情報系列については、式(2)が成り立つか否かを調べる必要はない。また、情報系列の長さを任意とするために、観測するブロック長 $L$ を無限大とすることは実現不可能であるので、 $L$ はある有限の長さであるものとする。更に、観測するブロック長 $L$ の範囲内において、状態が同一とならない場合は、これを無視するものとする。

#### 【0137】

ただし、ある符号変換表において、仮に、ある有限のブロック長 $L$ の任意の情報系列について式(2)が成り立っていた場合に、無限大のブロック長の任意の情報系列についても、式(2)が成り立つか否かについては、図18を用いて後述する。

#### 【0138】

例えば、図6の4状態符号変換表において、各情報語への符号語割当てを変更した場合、任意の情報系列について、式(2)を満たす符号語割当てが存在するか否かについて検証した結果、DC制御ビットを含む情報系列 $L$ の長さを4プロ

ック以上としたとき、DC制御ビットの挿入位置が、情報語の第1ビット目であっても、第2ビット目であっても、いずれの場合においても、図6の4状態符号変換表の、 $(4!)^4 = 331, 776$ 通りの、全ての符号語割当ての中に、任意の情報系列について、式(2)を満たす符号語割当ては1通りも存在しなかった。

#### 【0139】

また、同様に、図4の5状態符号変換表において、各情報語への符号語割当てを変更した場合、任意の情報系列について、式(2)を満たす符号語割当てが存在するか否かについて検証した結果、DC制御ビットを含む情報系列Lの長さを5ブロック以上としたとき、DC制御ビットの挿入位置が情報語の第1ビット目であっても、第2ビット目であっても、いずれにおいても、図4の5状態符号変換表の、 $(4!)^5 = 7, 962, 624$ 通りの符号語割当ての中に、任意の情報系列について式(2)を満たす符号語割当ては1通りも存在しなかった。

#### 【0140】

このように、従来知られている4状態もしくは5状態の有限状態型の(1, 7)RL符号の符号変換表においては、その符号語割当てを変化させたとしても、任意の情報系列について式(2)を満たす符号語割当ては1通りも存在しなかった。

#### 【0141】

一方、図7および図8の先見型の符号変換表において、各3ビット符号語を、3ブロック遅延させて、2ビット情報語に割り当てなおした104状態の有限状態型の符号変換表、並びに、図7、図8、および、図9の先見型の符号変換表において、各3ビット符号語を、5ブロック遅延させて、2ビット情報語に割り当てなおした1691状態の有限状態型の符号変換表については、いずれの場合においても、任意の情報系列について式(2)を満たしていた。しかしながら、まず、状態数のできるだけ少ない有限状態型の符号変換表を設計してから、任意の情報系列について式(2)を満たす符号語割当てを見つけるのであれば、パリティ保存型符号語割当てが行われた場合の符号よりも状態数を削減するのは、非常に困難である。



## 【0142】

このため、従来用いられていた近似固有ベクトルとは異なる近似固有ベクトルを用いて、ACHアルゴリズムによって新しい有限状態表を設計することによって、状態数の比較的少ない有限状態型の符号変換表において、任意の情報系列について式(2)が成立する符号語割当てを得ることができるようにした。

## 【0143】

すなわち、図5の有限状態遷移図から求められる8次元の近似固有ベクトルとして、新たに近似固有ベクトル[4, 6, 6, 6, 4, 4, 4, 2]を導出した。この近似固有ベクトルは、従来用いられている近似固有ベクトル[2, 3, 3, 3, 2, 2, 2, 1]の各要素を2倍したものである。この新しい近似固有ベクトルを基に、ACHアルゴリズムを用いて、新たに7状態の有限状態表を求めた。

## 【0144】

図15は、近似固有ベクトル[4, 6, 6, 6, 4, 4, 4, 2]を基に、ACHアルゴリズムを用いて求められた、(1, 7)RL制限を与える7状態の有限状態表である。図15の7状態有限状態表においては、まだ、情報語への符号語割当ては行われていない。図15の有限状態表から符号変換表を得るためには、各状態の3ビット符号語を、それぞれ、00、01、10、11の4種類の2ビット情報語に割り当てる必要がある。図15の有限状態表におけるこのような符号語割当ては、全部で $(4!)^7 = 4,586,471,424$ 通り存在する。

## 【0145】

次に、図15の有限状態表において、情報語への符号語割当てを行った際、任意の情報系列について式(2)を満たす符号語割当てが存在するか否かを求めるための処理について説明する。

## 【0146】

ある有限状態表において、観測するブロック長Lによらず任意の情報系列について式(2)を満たす符号語割当てを検索する処理は、例えば、図16に示されるパーソナルコンピュータなどを用いて実行することができる。

## 【0147】

図 1 6 は、パーソナルコンピュータ 1 2 1 の構成例を表している。パーソナルコンピュータ 1 2 1 の CPU (Central Processing Unit) 1 3 1 は、ROM (Read Only Memory) 1 3 2 に記憶されているプログラム、または、ハードディスク (HDD) 1 3 8 から RAM (Random Access Memory) 1 3 3 にロードされたプログラムに従って各種の処理を実行する。RAM 1 3 3 にはまた、CPU 1 3 1 が各種の処理を実行する上において必要なデータなども適宜記憶される。

#### 【 0 1 4 8 】

CPU 1 3 1、ROM 1 3 2、および RAM 1 3 3 は、内部バス 1 3 4 を介して相互に接続されている。この内部バス 1 3 4 にはまた、入出力インタフェース 1 3 5 も接続されている。

#### 【 0 1 4 9 】

入出力インタフェース 1 3 5 には、キーボード、マウスなどよりなる入力部 1 3 7、CRT (Cathode Ray Tube)、LCD (Liquid Crystal Display) などよりなり、画像やテキストを表示するディスプレイ、および、音声を出力するスピーカなどよりなる出力部 1 3 6、情報を記録再生する HDD 1 3 8、モデム、ターミナルアダプタなどより構成されるネットワークインターフェース 1 4 0 が接続されている。ネットワークインターフェース 1 4 0 は、例えば、インターネットなどのネットワークを介しての通信処理を行う。

#### 【 0 1 5 0 】

入出力インタフェース 1 3 5 にはまた、必要に応じてドライブ 1 3 9 が接続され、磁気ディスク 1 4 1、光ディスク 1 4 2、光磁気ディスク 1 4 3、もしくは半導体メモリ 1 4 4 などが適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて HDD 1 3 8 にインストールされる。

#### 【 0 1 5 1 】

図 1 7 のフローチャートを用いて、図 1 6 を用いて説明したパーソナルコンピュータ 1 2 1 が実行する、ある有限状態表において、観測するブロック長  $L$  によらず、任意の情報系列について、式 (2) を満たす符号語割当てを検索する処理について説明する。

#### 【 0 1 5 2 】

ステップS31において、CPU131は、入力部137より入力されたユーザの操作や、HDD138に予め記憶されている情報を基に、例えば、ACHアルゴリズムを用いて、検索に用いる有限状態表を取得する。例えば、CPU131は、図5の有限状態遷移図から、8次元の近似固有ベクトル $[4, 6, 6, 6, 4, 4, 4, 2]$ を求め、ACHアルゴリズムを用いて、 $(1, 7)$  RLL制限を与える7状態の有限状態表である、図15の7状態有限状態表を求める。

#### 【0153】

ステップS32において、CPU131は、RAM133に一時保存している変数Lおよび変数 $N_0$ を読み出して、それぞれ0を代入して初期化し、再びRAM133に保存する。ここで、変数Lは観測するブロック長で、変数 $N_0$ はブロック長を示す変数Lの任意の情報系列について、式(2)を満たす符号語割当ての個数である。

#### 【0154】

ステップS33において、CPU131は、RAM133に一時保存している変数Lの値を1インクリメントして $L=L+1$ とし、変数Mを $M=0$ とし、変数 $N_1$ に $N_0$ を代入して $N_1=N_0$ とした後、 $N_0$ に0を代入して、再びRAM133に保存する。ここで、変数 $N_1$ は、前回計算した $N_0$ の値であり、変数Mは符号語割当ての検索回数である。

#### 【0155】

ステップS34において、CPU131は、RAM133に一時保存している変数Mの値を1インクリメントして $M=M+1$ として、再びRAM133に保存し、符号語割当てを、変更して更新する。例えば、ステップS31において、図15の有限状態表を取得した場合、ステップS34において、4種類の2ビット情報語に対する3ビット符号語の割当て、すなわち、符号語割当てが、それまでに検索していない新しい符号語割当てに変更されて更新される。ただし、初回検索時の符号語割当ては任意でよく、2回目以降に、それまでに検索していない新しい符号語割当てを用いるようにすればよい。

#### 【0156】

ステップS35において、CPU131は、図12乃至図14を用いて説明し

たように、仮符号系列  $c_0$  および仮符号系列  $c_1$  において、状態  $s_0$  および状態  $s_1$  が一致した時点で、式 (2) を満たさない、すなわち、 $q_0 = q_1$  となるブロック長  $L$  の情報系列が 1 つでも存在するか否かを判断する。ステップ S 35 において、 $q_0 = q_1$  となるブロック長  $L$  の情報系列が 1 つも存在しないと判断された場合、処理は、ステップ S 36 に進む。ステップ S 35 において、 $q_0 = q_1$  となるブロック長  $L$  の情報系列が存在すると判断された場合、処理は、ステップ S 37 に進む。

#### 【0157】

ただし、ステップ S 35 において、2 つの仮符号系列の状態が一致する情報系列が 1 つも存在しない場合には、処理は、ステップ S 36 に進んでも、ステップ S 37 に進んでもかまわない。ここでは、この場合、ステップ S 37 に進むこととする。ただし、この場合、検索した全ての符号語割当ての中の少なくとも 1 つにおいて、2 つの仮符号系列の状態が一致する情報系列が少なくとも 1 つ存在すると仮定して、処理を実行するものとした。

#### 【0158】

ステップ S 35 において、 $q_0 = q_1$  となるブロック長  $L$  の情報系列が 1 つも存在しないと判断された場合、ステップ S 36 において、CPU 131 は、変数  $N_0$  の値を 1 インクリメントして、 $N_0 = N_0 + 1$  とするとともに、そのときの符号語割当てを用いた符号変換表、もしくは、そのときの符号語割当てを用いた符号変換表を表現するために必要十分な情報を、RAM 133 に保存する。

#### 【0159】

ステップ S 37 において、CPU 131 は、RAM 133 に一時保存している変数  $M$  と、符号語割当ての全数  $M_p$  とを読み出して比較し、 $M < M_p$  であるか否かを判断する。符号語割当ての全数  $M_p$  は、例えば、図 15 を用いて説明した有限状態表の場合、 $(4!)^7 = 4,586,471,424$  である。ステップ S 37 において、 $M < M_p$  であると判断された場合、処理は、ステップ S 34 に戻り、それ以降の処理が繰り返される。

#### 【0160】

ステップ S 37 において、 $M < M_p$  ではないと判断された場合、ステップ S 3

8において、CPU131は、RAM133に一時保存している変数 $N_1$ および $N_0$ を読み出し、 $N_0 \neq N_1$ かつ $N_0 \neq 0$ であるか否かを判断する。ステップS38において、 $N_0 \neq N_1$ かつ $N_0 \neq 0$ であると判断された場合、処理は、ステップS33に戻り、それ以降の処理が繰り返される。ステップS38において、 $N_0 \neq N_1$ かつ $N_0 \neq 0$ ではないと判断された場合、処理が終了される。

#### 【0161】

図17を用いて説明した検索処理の実行が終了した時点で、最後のLを用いた検索で保存された符号変換表が、観測するブロック長Lによらず任意の情報系列について式(2)を満たす全ての符号語割当てであり、 $N_0$ がその個数を表す。すなわち、 $N_0 = 0$ であれば、その有限状態表において、任意の情報系列について式(2)を満たす符号語割当ては、1通りも存在しない。

#### 【0162】

次に、図17のフローチャートに従って、図15の7状態の有限状態表において、全ての符号語割当てである $(4!)^7 = 4,586,471,424$ 通りについて、DC制御ビットを含む情報系列のブロック長Lを変化させた場合、任意の情報系列について式(2)が成立する符号語割当ての個数 $N(L)$ を求めた。

#### 【0163】

ただし、図15の有限状態表はACHアルゴリズムから作成されているため、図15の有限状態表においてどのような符号語割当てを用いようと、符号語割当て後の符号変換表に従って符号化された符号は、スライディングブロック復号されることが可能であるが、その復号時の符号語拘束長には、4ブロックのものと、3ブロックのものとが存在する。復号時の符号語拘束長が3ブロックとなる符号語割当ては、全体の $1/12$ に相当する382,205,952通りである。

#### 【0164】

図18に、図15の有限状態表において、観測するブロック長Lを変化させた場合の、ブロック長Lの任意の情報系列について式(2)を満足する符号語割当ての個数 $N(L)$ を、復号時の符号語拘束長が4ブロックのものと3ブロックのものとを分類して示す。また、DC制御ビットの挿入位置を、情報語の第1ビット目に固定しても第2ビット目に固定しても、式(2)を満足する符号語割当て

の個数  $N(L)$  の計算結果は同一である。

#### 【0 1 6 5】

すなわち、拘束長 4 では、ブロック長  $L$  が 1 のとき、955,514,880通りの符号語割当てで、式 (2) が満たされ、以下、ブロック数が増えるにしたがって、式 (2) を満たす符号語割当ての個数  $N(L)$  が減少し、ブロック長  $L$  が 6 のとき、327,680通りとなる。そして、ブロック長  $L$  が 7 となっても、符号語割当ての個数  $N(L)$  は、ブロック長  $L$  が 6 のときと同一であり、327,680通りとなる。

#### 【0 1 6 6】

一方、拘束長 3 では、ブロック長  $L$  が 1 のとき、63,700,992通りの符号語割当てで、式 (2) が満たされ、ブロック長  $L$  が 2 のとき、68,950,144通りの符号語割当てで、式 (2) が満たされる。以下、ブロック数が増えるにしたがって、式 (2) を満たす符号語割当ての個数  $N(L)$  が減少し、ブロック長  $L$  が 6 のとき、65,536通りとなる。そして、ブロック長  $L$  が 7 となっても、符号語割当ての個数  $N(L)$  は、ブロック長  $L$  が 6 のときと同一であり、65,536通りとなる。

#### 【0 1 6 7】

すなわち、図 1 8 において示されるように、式 (2) を満たす符号語割当ての個数  $N(L)$  は、拘束長 3 においてブロック長  $L$  が 1 から 2 に増える場合を除いて、ブロック長  $L$  が 6 以下の場合は、ブロック長が増えるにしたがって減少するが、ブロック長  $L$  が 6 から 7 に増える場合においては減少しない。

#### 【0 1 6 8】

また、ブロック長  $L$  が 6 である場合の任意の情報系列で式 (2) を満たす 393,216個 (拘束長 4 および拘束長 3) の符号語割当ては、全て、ブロック長  $L$  を 7 もしくは 8 に増加させた場合においても、任意の情報系列について、式 (2) を満たす。更に、この 393,216個の符号語割当てのうちのいくつかを、無作為に抽出した符号語割当ては、例えば、ブロック長  $L$  を 1 2 まで増加させた場合においても、任意の情報系列について、式 (2) を満たした。

#### 【0 1 6 9】

これは、ブロック長  $L$  をある程度大きくすると、符号状態の一致しない 2 つの仮符号系列が、全て繰り返しパターンとなるためである。したがって、仮にプロ

ック長  $L$  を無限大としても（実際には、ブロック長  $L$  を無限大とすることは不可能であるが）、この 393,216 個の符号語割当ては、任意の情報系列について、式（2）を満たす。

#### 【0170】

すなわち、観測するブロック長  $L$  を変化させて、式（2）を満たす符号語割当ての個数  $N(L)$  を計算した場合、次の式（3）が成立した時点において発見された符号語割当ての個数  $N(L)$  が、観測するブロック長  $L$  によらず、任意の情報系列について式（2）を満たす符号語割当ての個数である。

#### 【0171】

$$N(L) = N(L-1) \quad \dots (3)$$

#### 【0172】

また、図 18 において、復号時の符号語拘束長が 3 ブロックの場合において、観測するブロック長  $L$  が 1 の場合より、ブロック長  $L$  が 2 の場合のほうが、符号語割当ての個数  $N(L)$  の値が小さくなっている。これは、入力情報系列の長さが極端に短いと、仮符号系列  $c_0$  の状態  $s_0$  と仮符号系列  $c_1$  の状態  $s_1$  とが一致する場合の数が 0 となってしまう符号語割当てが発生してしまう場合があるが、このような符号語割当てについては、式（2）が成り立たないと見なしているためである。

#### 【0173】

図 18 を用いて説明したように、図 15 の有限状態表において、DC 制御ビットの挿入位置を情報語の第 1 ビット目もしくは第 2 ビット目のいずれか一方に固定した場合、観測するブロック長によらず、任意の情報系列について、式（2）を満たす符号語割当ては、全部で 393,216 通り存在する。ただし、ここで、上述したように、DC 制御ビットの挿入位置を、情報語の第 1 ビット目に固定しても第 2 ビット目に固定しても、任意の情報系列について式（2）を満たす符号語割当ての数は同一であるが、その符号語割当ては、必ずしも同一にはならない。

#### 【0174】

DC 制御ビットの挿入位置を、情報語の第 1 ビット目と第 2 ビット目のいずれの位置に挿入しても、任意の情報系列について式（2）を満たす符号語割当ての

数は、上述した393,216通りのうちの、ちょうど  $1/2$  に相当する、196,608通りである。

#### 【0175】

したがって、DC制御ビットの挿入位置を、情報語の第1ビット目もしくは第2ビット目の少なくともどちらか一方に挿入したときに、任意の情報系列について式(2)を満たす符号語割当ての数は、 $3 \times 196,608 = 589,824$ 通り存在する。この589,824個が、図15の有限状態表において、観測するブロック長によらず式(2)を満たす符号語割当ての全数である。この符号語割当て数は、図15の有限状態表における符号語割当ての全数の  $1/7776$  である。

#### 【0176】

ここで、図15の有限状態表において、観測するブロック長によらず、式(2)を満たす589,824通りの符号語割当てを、任意の情報系列について式(2)を満たす場合のDC制御ビットの挿入位置によって、3つのタイプに分類する。その3つのタイプとは、「情報語の第1ビット目にDC制御ビットを挿入したときのみ、任意の情報系列について式(2)を満たす符号語割当て」であるタイプ1、「情報語の第2ビット目にDC制御ビットを挿入したときのみ、任意の情報系列について式(2)を満たす符号語割当て」であるタイプ2、「情報語の第1ビット目または第2ビット目のいずれの位置にDC制御ビットを挿入しても、任意の情報系列について式(2)を満たす符号語割当て」であるタイプ3である。

#### 【0177】

図19に、タイプ1乃至タイプ3の、復号時の符号語拘束長が3ブロックである場合と4ブロックである場合との符号語割当ての数を示す。図19に示されるように、拘束長4ブロックにおける場合の符号語割当ての数は、拘束長3ブロックにおける場合の5倍であり、タイプ1乃至タイプ3では、それぞれ同数となる。

#### 【0178】

次に、図15の有限状態表に対して、具体的な符号語割当てを行った場合の符号変換表について、図20乃至図22を用いて、比較的良く似た符号語割当てと



なる 3 つの符号変換表について説明する。

【0179】

図 20 の符号変換表は、(1, 7) RLL 符号の有限状態型の 7 状態符号変換表の第 1 の例である。

【0180】

図 20 の符号変換表の符号語割当ては、復号時の符号語拘束長が 3 ブロックで、かつ、DC 制御ビットの挿入位置を情報語の第 1 ビット目と第 2 ビット目のいずれの位置に挿入しても、任意の情報系列について式 (2) を満たす符号語割当てであり、この符号語割当ては、図 19 に示されるタイプ 3 の 32,768 通りの符号語割当てのうちの 1 つである。

【0181】

図 21 の符号変換表は、(1, 7) RLL 符号の有限状態型の 7 状態符号変換表の第 2 の例である。

【0182】

図 21 の符号変換表の符号語割当ては、図 20 の符号変換表と同様に、復号時の符号語拘束長が 3 ブロックで、かつ、DC 制御ビットの挿入位置を情報語の第 1 ビット目と第 2 ビット目のいずれの位置に挿入しても、任意の情報系列について式 (2) を満たす符号語割当てであり、この符号語割当ては、図 19 に示されたタイプ 3 の 32,768 通りの符号語割当てのうちの 1 つである。図 21 の符号変換表は、図 20 の符号変換表とは異なる特徴を有しており、この点については後述する。

【0183】

図 22 の符号変換表は、(1, 7) RLL 符号の有限状態型の 7 状態符号変換表の第 3 の例である。

【0184】

図 22 の符号変換表の符号語割当ては、復号時の符号語拘束長が 3 ブロックで、かつ、DC 制御ビットの挿入位置を情報語の第 1 ビット目に挿入したときのみ、任意の情報系列について式 (2) を満たす符号語割当てであり、この符号語割当ては、図 19 に示されたタイプ 1 の 32,768 通りの符号語割当てのうちの 1 つ

である。図 22 の符号変換表は、図 20 および図 21 に示される符号変換表とは異なる特徴を有しており、この点については後述する。

#### 【0185】

ここで、図 23 を用いて、図 21 の符号変換表が、式 (2) を満たす具体的な例として、情報系列  $[1, 1, 0, 0, 0, 1, 0]$  の 7 ビットが入力された場合について説明する。

#### 【0186】

情報系列として  $[1, 1, 0, 0, 0, 1, 0]$  の 7 ビットが供給された場合、その先頭に仮の DC 制御ビットを挿入した情報系列  $d_0$  および情報系列  $d_1$ 、状態 3 を始点として、図 21 の符号変換表を用いて符号化された仮符号系列  $c_0$  および仮符号系列  $c_1$ 、並びに、符号化時の符号状態  $s_0$  および符号状態  $s_1$  は、図 23 に示されるように、情報系列  $d_0$  : 01, 10, 00, 10、符号系列  $c_0$  : 000, 100, 001, 001、符号状態  $s_0$  : 1, 6, 6, 3、情報系列  $d_1$  : 11, 10, 00, 10、符号系列  $c_1$  : 000, 010, 101, 001、符号状態  $s_1$  : 4, 7, 6, 3 となる。

#### 【0187】

この例では、3 ブロック目で仮符号系列  $c_0$  および仮符号系列  $c_1$  の符号状態  $s_0$  および符号状態  $s_1$  は、 $s_0 = s_1 = 6$  で同一となっており、符号状態が同一となった時点までの符号系列  $c_0 = [0, 0, 0, 1, 0, 0, 0, 0, 1]$  の合計の 2 の補数  $q_0$  は 0、符号系列  $c_1 = [0, 0, 0, 0, 1, 0, 1, 0, 1]$  の合計の 2 の補数  $q_1$  は 1 である。したがって、この場合、式 (2) が成立している。

#### 【0188】

また、図 21 の符号変換表を用いた符号変換においては、 $[1, 1, 0, 0, 0, 1, 0]$  の 7 ビットとは異なる情報系列を符号変換する場合や、DC 制御ビットの挿入位置を、情報語の第 2 ビット目とした場合においても、上述した 2 つの仮符号系列の状態が一致したとき、同様に式 (2) が成立する。

#### 【0189】

図 20 乃至図 22 の有限状態型の符号変換表は、それぞれ異なる特徴を有して

いる。しかしながら、図 20 乃至図 22 の、有限状態型の符号変換表からは、それぞれの特徴の差異は分かりにくい。

#### 【0190】

次に、図 20 乃至図 22 の有限状態型の符号変換表を、先見型の符号変換表へ変換する処理、並びに、それぞれの先見型符号変換表の特徴、および、その差異について説明する。

#### 【0191】

上述したように、先見型の符号変換表を有限状態型の符号変換表に変換することについては、従来実行されてきたが、その逆変換が行われた例はない。有限状態型の符号変換表を先見型の符号変換表へ変換するには、符号化時に必要な先見量分進んだ符号語を情報語に割り当てなおせば良いが、これに対して、有限状態型の符号変換表しか与えられていない場合、必要な先見量を符号変換表から知ることが必ずしも容易ではないため、有限状態型の符号変換表を先見型の符号変換表へ変換する処理は、先見型の符号変換表を有限状態型の符号変換表へ変換する処理と比較して困難である。

#### 【0192】

例えば、有限状態型の符号変換表が ACH アルゴリズムによって設計された場合、先見量の最大値は、用いられた近似固有ベクトルの各要素の最大値から 1 を引いた値となることが一般的に知られている。しかし、この値は、先見量の最大値であるため、必要な最小限の先見量とは、必ずしも一致しない。

#### 【0193】

ここで、従来から符号化に用いられていた可変長符号において、復号処理に必要な先見量を  $L_d$  ブロックとしたとき、符号化処理に必要な先見量は、 $L_d$  が偶数のとき、 $L_d/2$ 、 $L_d$  が奇数のとき、 $(L_d-1)/2$  である。復号処理において必要な先見量は、有限状態型の符号変換表を基に復号処理時の符号語拘束長を調べる際に知ることができる。

#### 【0194】

例えば、図 20 乃至図 22 の符号変換表に従って符号化された符号においては、復号時の符号語拘束長は、現時刻分を含めて 3 ブロックであり、復号時に必要

な先見量は2ブロックであることから、符号化において必要な先見量は1ブロックであると推定できる。

#### 【0195】

このことを利用して、図20乃至図22の符号変換表において、1ブロック進んだ符号語を、情報語に割り当てなおすことにより、これらの有限状態型の符号変換表を先見型の符号変換表に変換することができる。

#### 【0196】

なお、1ブロック進んだ符号語を、情報語に割り当てなおすとは、例えば、図21において、状態4は直前の情報語が必ず11となっていることから、状態4から発生される符号語010を、情報語11に割り当てるといった操作を行うことである。

#### 【0197】

図24は、1ブロック進んだ符号語を、情報語に割り当てなおすことにより、図20の有限状態型の符号変換表から変換された、(1, 7) RLL符号の先見型の符号変換表である。

#### 【0198】

図25は、1ブロック進んだ符号語を、情報語に割り当てなおすことにより、図21の有限状態型の符号変換表から変換された、(1, 7) RLL符号の先見型の符号変換表である。

#### 【0199】

図26は、1ブロック進んだ符号語を、情報語に割り当てなおすことにより、図22の有限状態型の符号変換表から変換された、(1, 7) RLL符号の先見型の符号変換表である。

#### 【0200】

図24乃至図26の先見型の符号変換表においては、1列目が、供給された情報ビットを示し、2列目が、与えられる符号を示す。また、(“0”)は、直前の符号ビットが0である場合を示し、(“1”)は、直前の符号ビットが1である場合を示す。

#### 【0201】

次に、図 24 乃至図 26 の先見型の符号変換表のそれぞれの特徴について説明する。

#### 【0202】

図 24 の先見型の符号変換表は、情報系列のパリティと符号系列のパリティとが常に一致する、式 (1) で表現されるパリティ保存型符号語割当てとなっている。ただし、図 24 によって表される (1, 7) RLL 符号は、従来の図 7 および図 8 によって表される符号と比較して、符号化時の状態数が 104 から 7、復号時の符号語拘束長が 5 ブロックから 3 ブロック、復号時の誤り伝播長が 10 ビットから 6 ビットと、それぞれ大幅に改善されている。図 24 に示される符号変換表と同様の特徴を有する符号語割当ては、図 19 に示されるタイプ 3 の符号語割当ての数のうちの 1/2 存在する。

#### 【0203】

図 25 の先見型の符号変換は、ブロック数が奇数の場合、情報系列のパリティと符号系列のパリティとが異なるように、かつ、ブロック数が偶数の場合、情報系列のパリティと符号系列のパリティとが一致するように符号語割当てが行われており、明らかに、式 (1) で表現されるパリティ保存型符号語割当てとは異なる符号語割当てが行われている。

#### 【0204】

しかしながら、先見型の符号変換表において、図 25 のような符号語割当ての符号は、図 24 を用いて説明した、パリティ保存型符号語割当ての符号と、同等の DC 制御が可能である。なぜならば、図 25 のような符号語割当てを用いた場合、ブロック数が奇数であろうと偶数であろうと、DC 制御ビット挿入時の符号系列における極性反転が保証されるからである。図 25 に示される符号変換表と同様の特徴を有する符号語割当ては、図 19 に示されるタイプ 3 の符号語割当てのうちの 1/2 存在する。

#### 【0205】

そして、図 26 の先見型の符号変換表は、明らかにパリティ保存型符号語割当てと異なり、更に、図 25 のような符号語割当てとも異なっている。図 26 の符号変換表は、情報語 01 または 11 を奇数ブロック含む情報系列に対しては、情

報系列のパリティと符号系列のパリティとが異なるように、かつ、情報語 01 または 11 を偶数ブロック含む情報系列に対しては、情報系列のパリティと符号系列のパリティとが一致するように、符号語割当てが行われている。

#### 【0206】

図 26 の符号変換表を用いた場合に、DC 制御を効率良く行うことができるか否かは、一見して判断し難いが、図 26 のような符号語割当てを有した符号においても、DC 制御ビットの情報語への挿入位置を第 1 ビット目に固定することにより、図 24 や図 25 を用いて説明した符号語割当ての符号と同等の DC 制御が可能である。

#### 【0207】

以上説明したように、先見型の符号変換表において、式 (1) の条件を満たすには、情報系列のパリティと符号系列のパリティとが一致している必要があった。これに対して、有限状態型の符号変換表のための式 (2) の条件を満たすには、情報系列のパリティと符号系列のパリティとは必ずしも一致している必要がなく、2 つの仮符号系列のパリティが異なっていれば良い。すなわち、式 (2) の条件は、式 (1) の条件を含んだ条件となっている。

#### 【0208】

すなわち、図 17 のフローチャートに従って検索された有限状態型の符号変換表の符号語割当てには、式 (1) の条件を満たすもののみならず、先見型の符号変換表に変換した場合に、図 25 や図 26 を用いて説明した、従来のパリティ保存型符号語割当てとは異なる符号語割当てとなるものが、数多く含まれる。

#### 【0209】

以下においては、有限状態型の符号変換表において、任意の情報系列について、2 つの仮符号系列の状態が一致した場合、必ず式 (2) が成り立つような符号語割当てを、「パリティ相違型符号語割当て」と称し、このような符号語割当ての方法を、「パリティ相違型符号語割当て法」と称するものとする。

#### 【0210】

次に、図 11 乃至図 13 を用いて説明した符号化部 51 が、図 4 および図 6、並びに、図 20 乃至図 22 を用いて説明した、有限状態型の各符号変換表に従っ

て、情報系列の符号化を行い、そのDC制御冗長度を変化させた場合の、低周波数帯域における符号スペクトルの抑圧効果について、図27および図28を用いて説明する。

#### 【0211】

具体的には、符号の低周波数スペクトルの抑圧効果の指標として、チャネルクロックを $f_c$ としたとき、 $f_c/1024$ の電力スペクトル密度を、その代表値として用いるものとする。また、NRZI変調した後の符号系列4096ビットについて、離散フーリエ変換を施して、 $f_c/1024$ のスペクトル成分を求める操作を、異なるランダムな情報系列について10000回繰り返し、それらの自乗平均を測定した。

#### 【0212】

図27に、DC制御ビットの挿入位置を、情報語の第1ビット目として、図4および図6、並びに、図20乃至図22を用いて説明した符号変換表に従って符号化を実行した場合の、 $f_c/1024$ の電力スペクトル密度のDC制御冗長度依存性を示し、図28に、DC制御ビットの挿入位置を、情報語の第2ビット目として、図4および図6、並びに、図20乃至図22の符号変換表に従って符号化を実行した場合の、 $f_c/1024$ の電力スペクトル密度のDC制御冗長度依存性を示す。

#### 【0213】

図27に示されるように、DC制御ビットの挿入位置を情報語の第1ビット目として、図20乃至図22の符号変換表に基づいて符号化処理を実行した場合、DC制御冗長度が0.007乃至0.12の広い範囲において、従来の有限状態型の符号変換表に基づいて符号化した場合よりも、低周波数スペクトルの抑圧効果を改善することができる。

#### 【0214】

また、図28に示されるように、DC制御ビットの挿入位置を情報語の第2ビット目として、図20および図21の符号変換表に基づいて符号化処理した場合、DC制御冗長度が0.007乃至0.08の広い範囲において、従来の有限状態型の符号変換表に基づいて符号化した場合よりも、低周波数スペクトルの抑圧効果を改善することができる。

## 【0215】

更に、図27および図28に示されるように、図22の符号変換表を用いた場合、DC制御ビットの挿入位置を情報語の第1ビット目としたときには、高い低周波数スペクトルの抑圧効果を得ることができるが、情報語の第2ビット目としたときには、従来の4状態もしくは5状態の符号変換表を用いた場合よりも、得られる低周波数スペクトルの抑圧効果が低くなってしまう。これは、図22の符号変換表において、DC制御ビットの挿入位置を情報語の第1ビット目としたときのみ、任意の情報系列について式(2)を満たすように、符号語割当てが行われているためである。

## 【0216】

ただし、どのような符号変換表においても、情報語01に割り当てられた符号語と情報語10に割り当てられた符号語とを相互に全て入れ替えるか、もしくは情報語00に割り当てられた符号語と情報語11に割り当てられた符号語とを相互に全て入れ替えることにより、DC制御ビットの挿入位置を情報語の第1ビット目とした場合と情報語の第2ビット目とした場合とで、その性質が完全に入れ代わった符号語割当てとすることができる。

## 【0217】

すなわち、本発明を適用した符号化処理においては、DC制御ビットの挿入位置を、情報語の第1ビット目とするか第2ビット目とするかはあまり重要ではなく、挿入するDC制御ビットの挿入位置に合わせて低周波数帯域の抑圧効果を高くできる符号語割当てを選択するか、もしくは、用いられる符号語割当てに合わせて、DC制御ビットの挿入位置を情報語の第1ビット目とするか第2ビット目とするかを決定するようにすればよい。

## 【0218】

図27より、例えば、図21の7状態の符号変換表に従って符号化を実行した場合、図6の4状態の符号変換表に従って符号化した場合と比較して、 $-8.5\text{dB}$ の $f_c/1024$ での電力スペクトル密度を得るのに、DC制御ビットの冗長度を $1/22=0.04545$ から $1/40=0.025$ に大きく改善することができる。ただし、実際のシステムにおいて、 $f_c/1024$ での電力スペクトル密度を、どの程度まで抑圧す



る必要があるかは、用いるシステムによっても異なるが、一般的には、 $-6\text{dB}$ 以下である。また、符号スペクトルの低周波数帯域の抑圧効果は、DC制御の方法によっても異なる。

#### 【0219】

このように、DC制御ビットを情報系列に挿入した場合、図20乃至図22の有限状態型の符号変換表に従って符号化することにより、符号系列のDC制御を効率良く行うことが可能となる。

#### 【0220】

すなわち、符号系列のDC制御を行うためのDC制御ビットを情報系列に挿入した後、 $m$ ビットの情報語を $n$ ビットの符号語に連続して変換する符号化部51の処理において用いられる符号化規則の第1の例においては、符号化規則を有限状態型の符号変換表で表現したとき、符号変換表が、DC制御ビットを0と仮定して任意の始点状態から符号化した仮符号系列 $c_0$ の符号状態 $s_0$ と、DC制御ビットを1と仮定して、仮符号系列 $c_0$ と同一の始点状態から符号化した仮符号系列 $c_1$ の符号状態 $s_1$ とが同一となった時点で、仮符号系列 $c_0$ の合計の2の補数と、仮符号系列 $c_1$ の合計の2の補数とが、常に異なるように、情報語に符号語が割り当てられている。

#### 【0221】

また、符号系列のDC制御を行うためのDC制御ビットを情報系列に挿入した後、 $m$ ビットの情報語を $n$ ビットの符号語に連続して変換する符号化部51の処理において用いられる符号化規則の第2の例においては、第1の例の条件に加えて、更に、符号化規則を先見型の符号変換表で表現したとき、 $m$ ビットの情報語もしくは $n$ ビットの符号語を1ブロックとして、ブロック数が奇数のときには、情報系列の合計の2の補数と符号系列の合計の2の補数とが異なるように、かつ、ブロック数が偶数のときには、情報系列の合計の2の補数と符号系列の合計の2の補数とが一致するように、符号語が情報語に割り当てられている。

#### 【0222】

また、符号系列のDC制御を行うためのDC制御ビットを情報系列に挿入した後、 $m$ ビットの情報語を $n$ ビットの符号語に連続して変換する符号化部51の処

理において用いられる符号化規則の第3の例においては、符号化規則を有限状態型の符号変換表で表現したとき、2ビット情報語の第1ビット目もしくは第2ビット目のどちらか一方にDC制御ビットを挿入した場合のみに、符号変換表が、DC制御ビットを0と仮定して任意の始点状態から符号化した仮符号系列 $c_0$ の符号状態 $s_0$ と、DC制御ビットを1と仮定して、仮符号系列 $c_0$ と同一の始点状態から符号化した仮符号系列 $c_1$ の符号状態 $s_1$ とが同一となった時点で、仮符号系列 $c_0$ の合計の2の補数と、仮符号系列 $c_1$ の合計の2の補数とが、常に異なるように、情報語に符号語が割り当てられる。

### 【0223】

以上説明したように、有限状態型の符号変換表におけるパリティ相違型符号語割当て法は、先見型の符号変換表における従来のパリティ保存型符号語割当て法を含み、かつ、従来のパリティ保存型符号語割当て法以外の非常に多くのDC制御効率の良い符号語割当てを与えるものである。ただし、有限状態型の符号変換表におけるパリティ相違型符号語割当てが、結果的に、先見型の符号変換表におけるパリティ保存型符号語割当てとなった場合においても、パリティ相違型符号語割当ての方がより理論的な設計が可能のため、図7および図8を用いて説明した、従来のパリティ保存型符号語割当てを行った符号よりも、特性を大きく改善することが可能である。

### 【0224】

次に、(1, 7) RLL符号における最小ランの最大連続数を、従来の6よりも小さくし、再生処理時に、デフォーカスやタンジェンシャルチルトが発生した場合のビット誤り率を改善する場合について説明する。

### 【0225】

図29に、(1, 7) RLL符号における、最小ランの最大連続数と符号のシャノン容量 (Shannon capacity) との関係を示す。ここでシャノン容量とは、制限の与えられた符号の達成し得る理論上の最大符号化率である。上述したACHアルゴリズムを用いることにより、シャノン容量以下の符号化率の符号の有限状態型の符号変換表を設計することができる。

### 【0226】

図 29 より、(1, 7) RLL 符号における最小ランの最大連続数は、符号化率を  $2/3=0.6667$  のままとすれば、シャノン容量が 0.6730 である最小ラン最大連続数 3 まで低減させることが可能であることが分かる。

#### 【0227】

すなわち、2 ビットの情報語を 3 ビットの符号語に連続して変換し、かつ、符号の最小ラン制限が 1、最大ラン制限が 7 である符号化規則を用いて符号化を行う場合、符号の最小ランの最大連続数は、3 以上 5 以下に制限される。

#### 【0228】

次に、その具体例として、最小ランの最大連続数を 5 とした場合の、(1, 7) RLL 符号の、具体的な有限状態型の符号変換表の設計方法について説明する。

#### 【0229】

図 30 は、NRZI 変調を前提とした場合の、(1, 7) RLL の制限規則と、最小ランの最大連続数を 5 とする制限とを、同時に与える 18 状態の有限状態遷移図である。

#### 【0230】

ここで、図 30 において、最小ランの最大連続数を 4 としたい場合には、状態 17 乃至 18 を削除した状態遷移図を用い、3 としたい場合には、状態 15 乃至 18 を削除した状態遷移図を用いるようにすれば良い。

#### 【0231】

まず、図 30 の有限状態遷移図を基に、近似固有ベクトルとして [4, 6, 6, 6, 5, 5, 4, 2, 4, 6, 3, 6, 3, 5, 3, 5, 2, 3] が導出される。そして、この近似固有ベクトルを基に、ACH アルゴリズムに基づいて、図 15 の 7 状態の有限状態表と類似した、8 状態の有限状態表を設計することができる。図 31 に、最小ランの最大連続数を 5 とした (1, 7) RLL 符号のための有限状態表を示す。

#### 【0232】

そして、図 17 を用いて説明したフローチャートに基づいて、図 31 の有限状態表を用いて、その全ての符号語割当てである  $(4!)^8=110,075,314,176$  通り

の中から、観測するブロック長によらず、任意の情報系列について式(2)を満たす符号語割り当て、すなわちパリティ相違型符号語割り当てを探索した。その結果、DC制御ビットの挿入位置を情報語の第1ビット目もしくは第2ビット目のどちらか一方のみに固定した場合、 $L=7$ において、式(3)が成立した。そして、そのとき、パリティ相違型符号語割り当てとなる符号語割り当て数は、786,432通り存在した。また、DC制御ビットの挿入位置を情報語の第1ビット目もしくは第2ビット目の少なくともどちらか一方に挿入した場合、パリティ相違型符号語割り当てとなる符号語割り当て数は1,179,648通りであった。この符号語割り当て数は、図31の有限状態表における符号語割り当ての全数の $1/93312$ である。

#### 【0233】

ただし、図31の有限状態表は、ACHアルゴリズムを用いて作成されているため、図31の有限状態表において、どのような符号語割り当てを用いようと、符号語割り当て後の符号変換表にしたがって符号化された符号は、スライディングブロック復号されることが可能であるが、その復号時の符号語拘束長は5ブロックのものと4ブロックのものが存在する。更に、復号時の符号語拘束長が4ブロックのものには、2ビット情報語のうち、一方のビットのみの符号語拘束長が3ブロックのものが存在する。このような場合、その符号語拘束長が4ブロックであることに変わりはないが、復号時の誤り伝播長が1ビット短くなる。

#### 【0234】

図32を用いて、図31の有限状態表において、パリティ相違型符号語割り当てとなる1,179,648通りの符号語割り当て数の内訳について説明する。図32におけるタイプ分類は、図19を用いて説明した、タイプ1乃至タイプ3と同一の分類である。また、図32においては、復号時の符号語拘束長が、5ブロックの符号語割り当てと4ブロックの符号語割り当てとを分類して記述しており、更に、括弧内は、符号語拘束長が4ブロックの符号語割り当てのうち、2ビット情報語の一方のビットの符号語拘束長が3ブロックとなる符号語割り当て数である。

#### 【0235】

図33は、パリティ相違型符号語割り当てされ、最小ランの最大連続数が5に制

限された、(1, 7) RLL符号の有限状態型の8状態符号変換表の具体例である。図33の符号変換表の符号語割当ては、復号時の符号語拘束長が4ブロックで、かつ、2ビット情報語のうち、一方のビットの符号語拘束長が3ブロックであり、かつ情報語の第1ビット目および第2ビット目のどちらをDC制御ビットとしてもパリティ相違型符号語割当てとなる、図32に示されたタイプ3の256通りの符号語割当ての1つである。

#### 【0236】

図33の符号変換表に従った符号は、従来の図7、図8、および、図9の符号変換表に従った符号と比較して、最小ランの最大連続数が6から5に削減されているだけでなく、符号化時の状態数が1691から8、復号時の符号語拘束長が6ブロックから4ブロック、復号時の誤り伝播長が15ビットから7ビットと、それぞれ大幅に改善されている。

#### 【0237】

このように、2ビットの情報語を3ビットの符号語に連続して変換し、かつ、符号の最小ラン制限が1、最大ラン制限が7である符号化方法において、符号の最小ランの最大連続数を5に制限した符号は、最小ランの最大連続数が6であるものよりも、優れた特性を有するものである。図11乃至図13を用いて説明した符号化部51は、図13のステップS6の処理において、最小ランの最大連続数を5に制限した符号を用いて符号化処理を行うことにより、従来における場合よりも、符号化時の状態数を減少し、復号時の符号語拘束長を短縮し、誤り伝播長を短縮することが可能となる。

#### 【0238】

また、その符号化規則を有限状態型の符号変換表で表現した場合、その状態数を8とすることができる。

#### 【0239】

また、図33の有限状態型の符号変換表についても、1ブロック進んだ符号語を情報語に割り当てなおすことにより、先見型の符号変換表に変換することができる。その結果得られた先見型の符号変換表は、基本変換表が図25と全く同一であり、図25の基本変換表に、図34に示す違反時変換表を加えたものとなる

。

#### 【0 2 4 0】

すなわち、図 3 4 の違反時符号変換表は、図 2 5 の先見型の符号変換表に対して、最小ランの最大連続数を 5 に制限するために追加される違反時符号変換表である。図 3 4 の違反時符号変換表は、情報語 “1 0 . 1 1” が供給された場合、直前の符号語が (0 1 0) であれば、図 2 5 の符号変換表に従わずに、符号語 (0 0 0 . 0 0 1) に変換することを示す、違反時符号変換表である。

#### 【0 2 4 1】

図 3 4 の違反時符号変換表においては、図 2 5 の場合と同様に、ブロック数が偶数のときに情報系列のパリティと符号系列のパリティとが一致するように、符号語割当てが行われている。図 2 5 の先見型の基本符号変換表に、図 3 4 の先見型の違反時符号変換表を加えた符号変換規則に基づいて符号化を行うことにより、パリティ相違型符号語割当てを保ったまま最小ランの最大連続数を 5 に制限することができる。ただし、この符号化方法は、図 3 3 の有限状態型の符号変換表に基づいて、符号化を行うことと等価である。

#### 【0 2 4 2】

図 3 5 は、図 2 0 乃至図 2 2 に示した 7 状態の符号変換表と、図 3 3 に示した 8 状態の符号変換表とにおける、符号化時の状態数、復号時の符号語拘束長、および復号時の誤り伝播長を計算した結果である。

#### 【0 2 4 3】

図 2 0 乃至図 2 2 に示した 7 状態の符号変換表と、図 3 3 に示した 8 状態の符号変換表を用いた符号化方法は、図 3 5 と図 1 0 とを比較することにより明らかなように、従来の図 7、図 8、および、図 9 の符号変換表を用いた符号化方法と比較して、符号化時の状態数、復号時の符号語拘束長、および復号時の誤り伝播長を大きく改善することができるものである。

#### 【0 2 4 4】

なお、以上説明した符号化方法を用いた符号を復号するための復号方法には、どのような方法を用いても良い。その復号方法としては、例えば、図 3 5 において復号時の符号語拘束長が求められていることから明らかなように、一般的なス

ライディングブロック復号などを用いるようにしても良い。また、その他の復号方法としては、本発明の符号化方法における符号状態数が比較的少ないことを利用して、例えば、有限状態型の符号変換表に従ったトレリス線図を用いた軟判定復号などを用いるようにしても良い。

#### 【0245】

例えば、図21もしくは図33の符号変換表に従って符号化された符号に対して、スライディングブロック復号を行う場合、入力される4ブロック（12ビット）の符号語列を過去側から  $[b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}]$  と表現すれば、2ビット情報語  $[a_0, a_1]$  は、次の式（4）および式（5）のブール（Bool）代数論理式に従って、非常に簡単な回路で復号できる。

#### 【0246】

##### 【数1】

$$a_0 = b_2(b_6 + \overline{b_5}b_7) + b_4(b_6 + b_7 + \overline{x}b_8\overline{b_9}b_{10}) + b_3b_5 + \overline{b_2}b_3b_4(\overline{b_1} + \overline{x}b_5\overline{b_6}b_7) \quad \dots (4)$$

$$a_1 = b_4 + \overline{b_2}b_3(b_0 \oplus b_5) + \overline{b_5}b_6b_7(\overline{b_1} + b_3 + x) \quad \dots (5)$$

#### 【0247】

ただし、ここで、 $x=0$ の場合、図33もしくは図21によって表される両方の符号について、 $x=1$ の場合、図21によって表される符号について、それぞれ復号することができる。また、 $b_{11}$ は、スライディングブロック復号において使用しない。また、復号時においては、符号化時に挿入されたDC制御ビットは、通常、破棄される。

#### 【0248】

上述した一連の処理は、ソフトウェアにより実行することもできる。そのソフトウェアは、そのソフトウェアを構成するプログラムが、専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどに、記録媒体からインストールされる。

## 【0249】

上述した一連の処理を実行するプログラムを記録する記録媒体は、図16に示すように、コンピュータとは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク141（フレキシブルディスクを含む）、光ディスク142（CD-ROM（Compact Disk-Read Only Memory）、DVD（Digital Versatile Disk）を含む）、光磁気ディスク143（MD（Mini-Disk）（商標）を含む）、もしくは半導体メモリ144などよりなるパッケージメディアなどにより構成される。

## 【0250】

また、本明細書において、記録媒体に記録されるプログラムを記述するステップは、記載された順序に沿って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的もしくは個別に実行される処理をも含むものである。

## 【0251】

## 【発明の効果】

このように、本発明によれば、符号変換を行うことができる。特に、符号化規則を有限状態型の符号変換表で表現した場合、所定の始点状態から符号化された第1の仮符号系列の第1の符号状態と、所定の始点状態から符号化された第2の仮符号系列の第2の符号状態とが同一となった時点で、第1の仮符号系列を構成する符号化ビットの合計の2の補数と、第2の仮符号系列を構成する符号化ビットの合計の2の補数とが、常に異なるように、情報語に符号語が割り当てられるような符号化規則を用いて、符号変換を行うようにすることができる。

## 【0252】

また、他の本発明によれば、符号変換を行うことができる他、最小ラン制限が1、最大ラン制限が7であり、かつ、符号語の最小ランの最大連続数が3乃至5となる符号化規則に基づいて、符号化処理を実行するようすることができる。

## 【図面の簡単な説明】

## 【図1】

従来の記録再生装置の構成を示すブロック図である。



**【図 2】**

(1, 7) R L L 符号の先見型の符号変換表について説明する図である。

**【図 3】**

(1, 7) R L L 符号の先見型の符号変換表について説明する図である。

**【図 4】**

図 2 および図 3 の符号変換表を変換した、有限状態型の 5 状態符号変換表について説明するための図である。

**【図 5】**

(1, 7) R L L 符号を与える 8 状態の状態遷移図である。

**【図 6】**

(1, 7) R L L 符号の有限状態型の 4 状態符号変換表について説明するための図である。

**【図 7】**

(1, 7) R L L 符号のパリティ保存型符号語割当てを用いた先見型の符号変換表について説明する図である。

**【図 8】**

(1, 7) R L L 符号のパリティ保存型符号語割当てを用いた先見型の符号変換表について説明する図である。

**【図 9】**

図 7 および図 8 の符号変換表に追加される、最小ランの最大連続数を 6 に制限するための、違反時符号変換表について説明する図である。

**【図 10】**

従来の (1, 7) R L L 符号の状態数、符号語拘束長、および誤り伝播長について説明する図である。

**【図 11】**

本発明を適用した記録再生装置の構成を示すブロック図である。

**【図 12】**

図 11 の符号化部の構成を示すブロック図である。

**【図 13】**

図 1 2 の符号化部が実行する符号化処理について説明するフローチャートである。

【図 1 4】

図 1 2 の符号化部が実行する符号化処理における、2 系列の状態について説明するための図である。

【図 1 5】

(1, 7) R L L 符号の 7 状態有限状態表について説明するための図である。

【図 1 6】

パーソナルコンピュータの構成を示すブロック図である。

【図 1 7】

符号変換表検索処理について説明するフローチャートである。

【図 1 8】

ブロック長ごとの、条件を満たす符号語割当て数について説明する図である。

【図 1 9】

ブロック長によらず条件を満たす符号語割当てのタイプ分類について説明する図である。

【図 2 0】

条件を満たす、(1, 7) R L L 符号の有限状態型の 7 状態符号変換表の第 1 の例について説明するための図である。

【図 2 1】

条件を満たす、(1, 7) R L L 符号の有限状態型の 7 状態符号変換表の第 2 の例について説明するための図である。

【図 2 2】

条件を満たす、(1, 7) R L L 符号の有限状態型の 7 状態符号変換表の第 3 の例について説明するための図である。

【図 2 3】

図 1 2 の符号化部が実行する符号化処理における、2 系列の状態について説明するための図である。

【図 2 4】

図 20 の符号変換表から変換された、条件を満たす、(1, 7) RLL 符号の先見型の符号変換表について説明するための図である。

【図 25】

図 21 の符号変換表から変換された、条件を満たす、(1, 7) RLL 符号の先見型の符号変換表について説明するための図である。

【図 26】

図 22 の符号変換表から変換された、条件を満たす、(1, 7) RLL 符号の先見型の符号変換表について説明するための図である。

【図 27】

電力スペクトル密度の DC 制御冗長度依存性について説明する図である。

【図 28】

電力スペクトル密度の DC 制御冗長度依存性について説明する図である。

【図 29】

最小ランの最大連続数とシャノン容量について説明する図である。

【図 30】

最小ランの最大連続数を 5 に制限する、(1, 7) RLL 符号の 18 状態の状態遷移図である。

【図 31】

最小ランの最大連続数を 5 に制限する、(1, 7) RLL 符号の 8 状態有限状態表について説明するための図である。

【図 32】

ブロック長によらず条件を満たす符号語割当てのタイプ分類による符号語割当て数について説明する図である。

【図 33】

最小ランの最大連続数を 5 に制限する、(1, 7) RLL 符号の有限状態型の 8 状態符号変換表について説明するための図である。

【図 34】

最小ランの最大連続数を 5 に制限するために、図 25 の基本符号変換表に追加される、違反時符号変換表について説明するための図である。

## 【図 3 5】

図 2 0 乃至図 2 2 に示した 7 状態の符号変換表と、図 3 3 に示した 8 状態の符号変換表とにおける、符号化時の状態数、復号時の符号語拘束長、および復号時の誤り伝播長について説明するための図である。

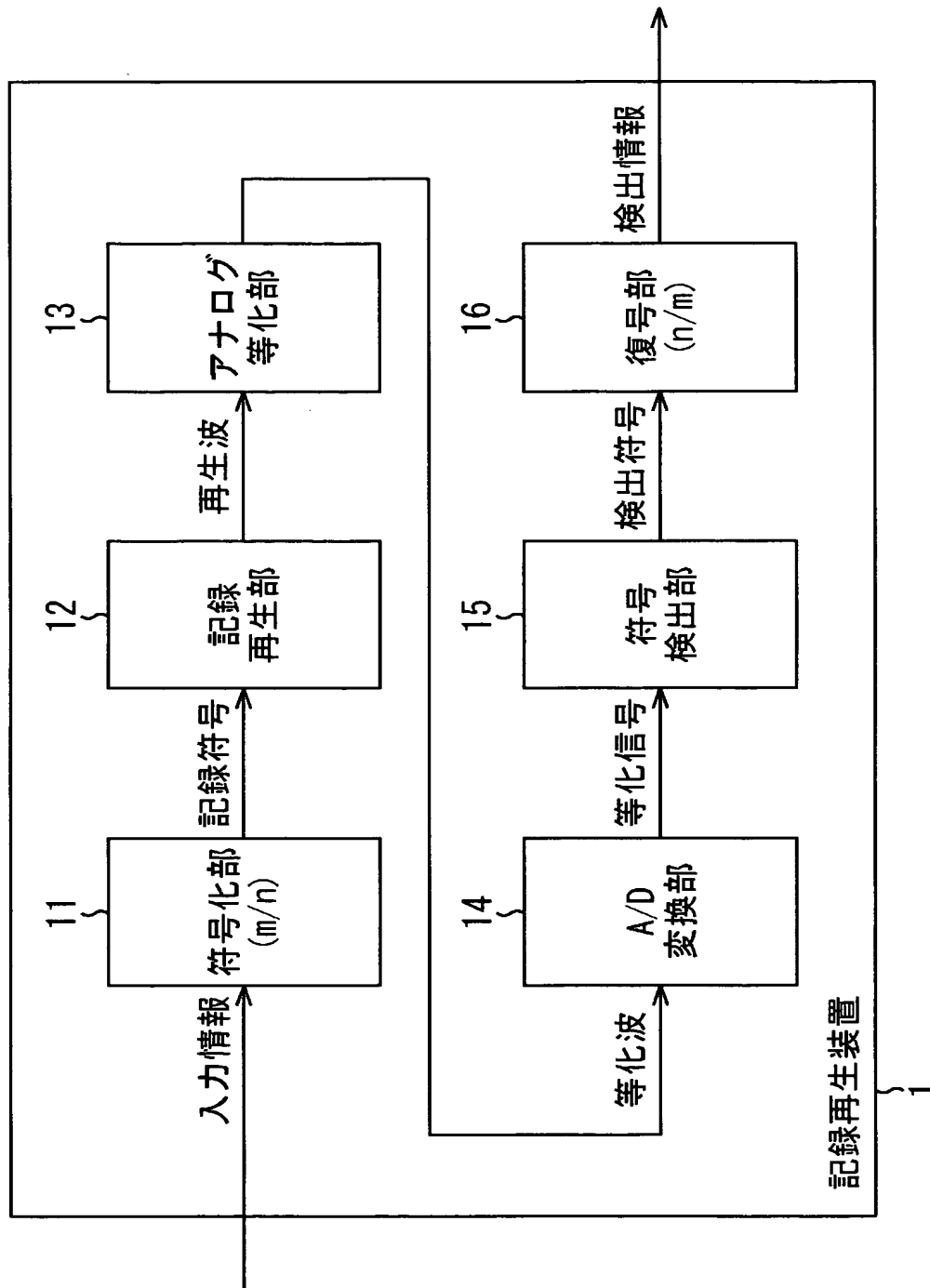
## 【符号の説明】

3 1 記録再生装置, 4 1 符号化装置, 5 1 符号化部, 1 0 1 D  
C 制御ビット挿入部, 1 0 2, 1 0 3  $m/n$  変換部, 1 0 6 符号選択信  
号生成部, 1 0 7 符号選択部

【書類名】 図面

【図 1】

図1



【図 2】

図2

| Data | Code |
|------|------|
| 00   | 101  |
| 01   | 100  |
| 10   | 001  |
| 11   | 010  |

【図 3】

図3

| Data  | Code    |
|-------|---------|
| 00.00 | 101.000 |
| 00.01 | 100.000 |
| 10.00 | 001.000 |
| 10.01 | 010.000 |

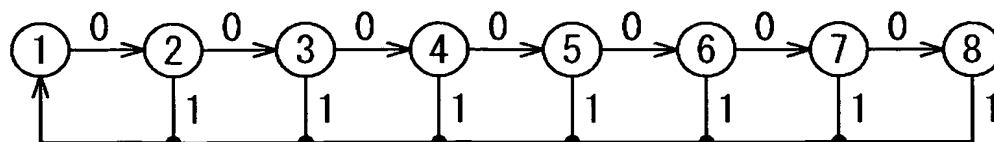
【図 4】

図4

|   | 00    | 01    | 10    | 11    |
|---|-------|-------|-------|-------|
| 1 | 101/5 | 100/5 | 101/3 | 101/4 |
| 2 | 100/1 | 100/2 | 100/3 | 100/4 |
| 3 | 001/5 | 010/5 | 001/3 | 001/4 |
| 4 | 010/1 | 010/2 | 010/3 | 010/4 |
| 5 | 000/1 | 000/2 | 000/3 | 000/4 |

【図 5】

図5



【図 6】

図6

|   | 00    | 01    | 10    | 11    |
|---|-------|-------|-------|-------|
| 1 | 010/1 | 010/2 | 010/3 | 100/3 |
| 2 | 100/1 | 100/2 | 101/3 | 101/4 |
| 3 | 000/1 | 000/2 | 001/3 | 001/4 |
| 4 | 010/1 | 010/2 | 010/3 | 000/3 |



【図 7】

図 7

| Data       | Code          |
|------------|---------------|
| ("0") 11   | 101           |
| ("1") 11   | 000           |
| 10         | 001           |
| 01         | 010           |
| 00. 11     | 010. 100      |
| 00. 10     | 010. 000      |
| 00. 01     | 000. 100      |
| 00. 00. 11 | 000. 100. 100 |
| 00. 00. 10 | 000. 100. 000 |
| 00. 00. 01 | 010. 100. 100 |
| 00. 00. 00 | 010. 100. 000 |

[("0")は直前の符号ビットが0]

[("1")は直前の符号ビットが1]

【図 8】

図 8

| Data           | Code               |
|----------------|--------------------|
| 00. 00. 10. 00 | 000. 100. 100. 100 |
| 00. 00. 00. 00 | 010. 100. 100. 100 |

【図 9】

図9

| Data               | Code          |
|--------------------|---------------|
| 11. 01. 11 ("010") | 001. 000. 000 |

[("010")は直後の符号語が010]

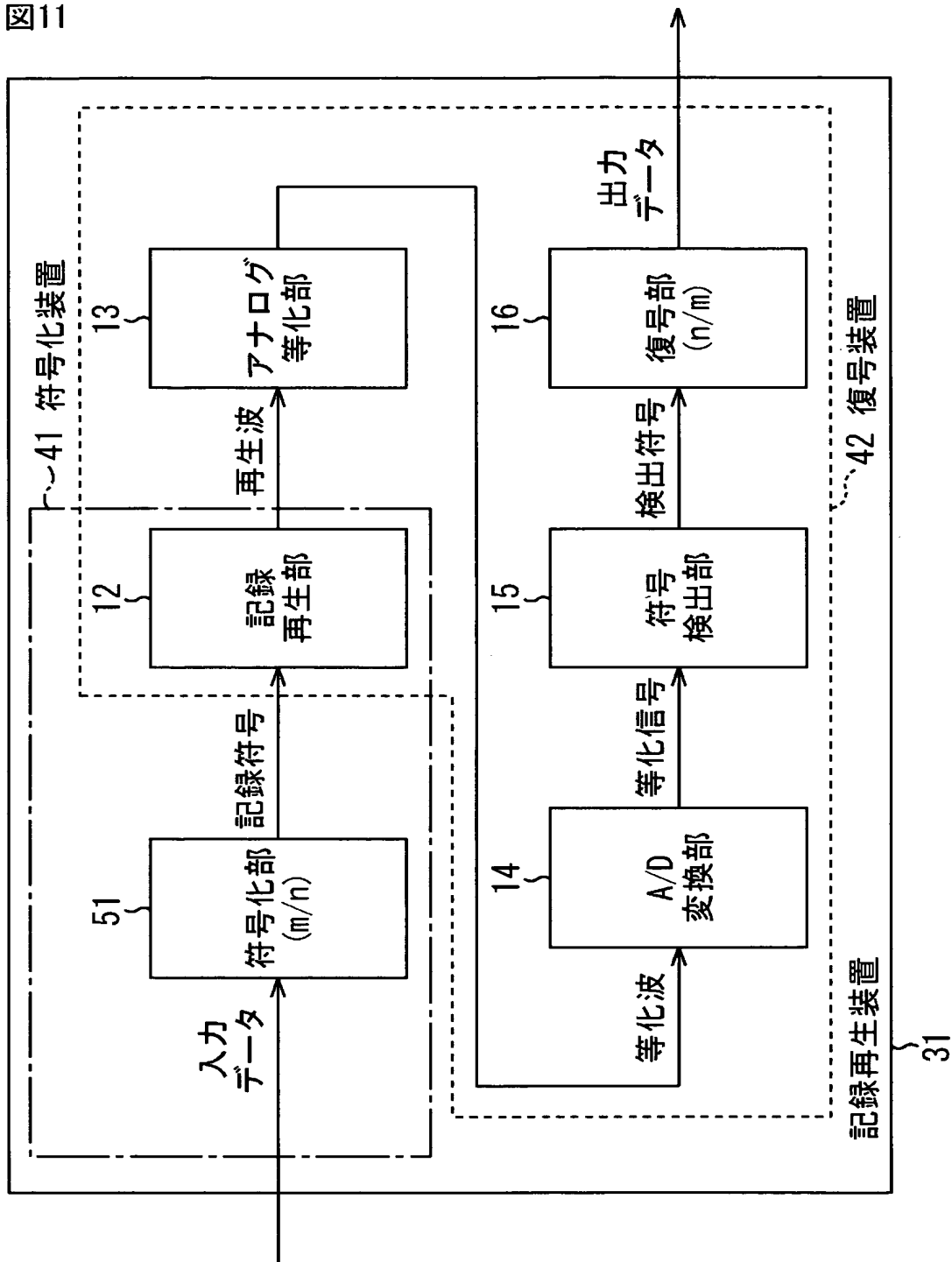
【図 1 0】

図10

| 符号変換表        | 状態数  | 符号語拘束長<br>(ブロック) | 誤り伝播長<br>(ビット) |
|--------------|------|------------------|----------------|
| 図2および図3または図4 | 5    | 3                | 5              |
| 図6           | 4    | 3                | 5              |
| 図7および図8      | 104  | 5                | 10             |
| 図7、図8、および図9  | 1691 | 6                | 11             |

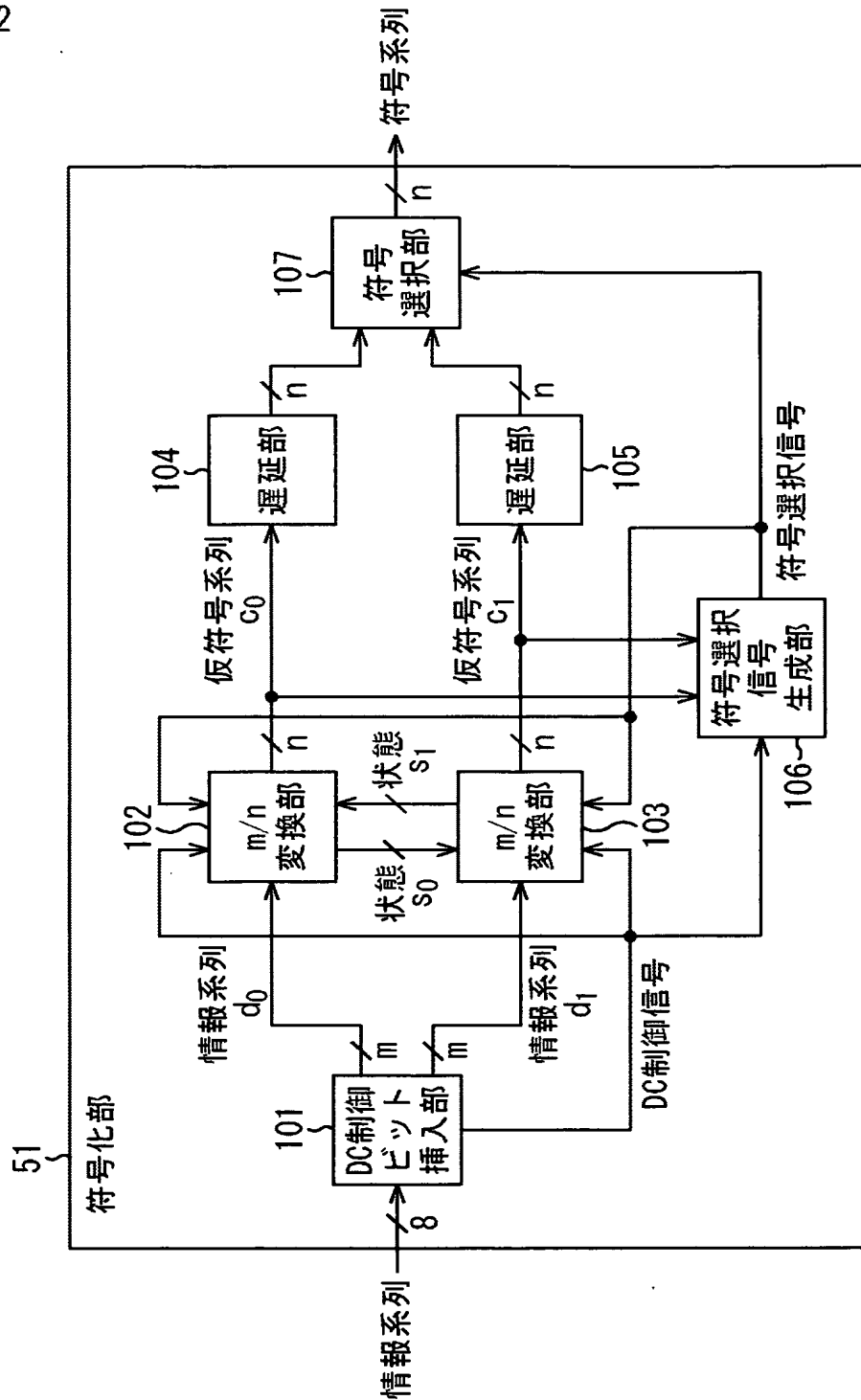
【図 11】

図11



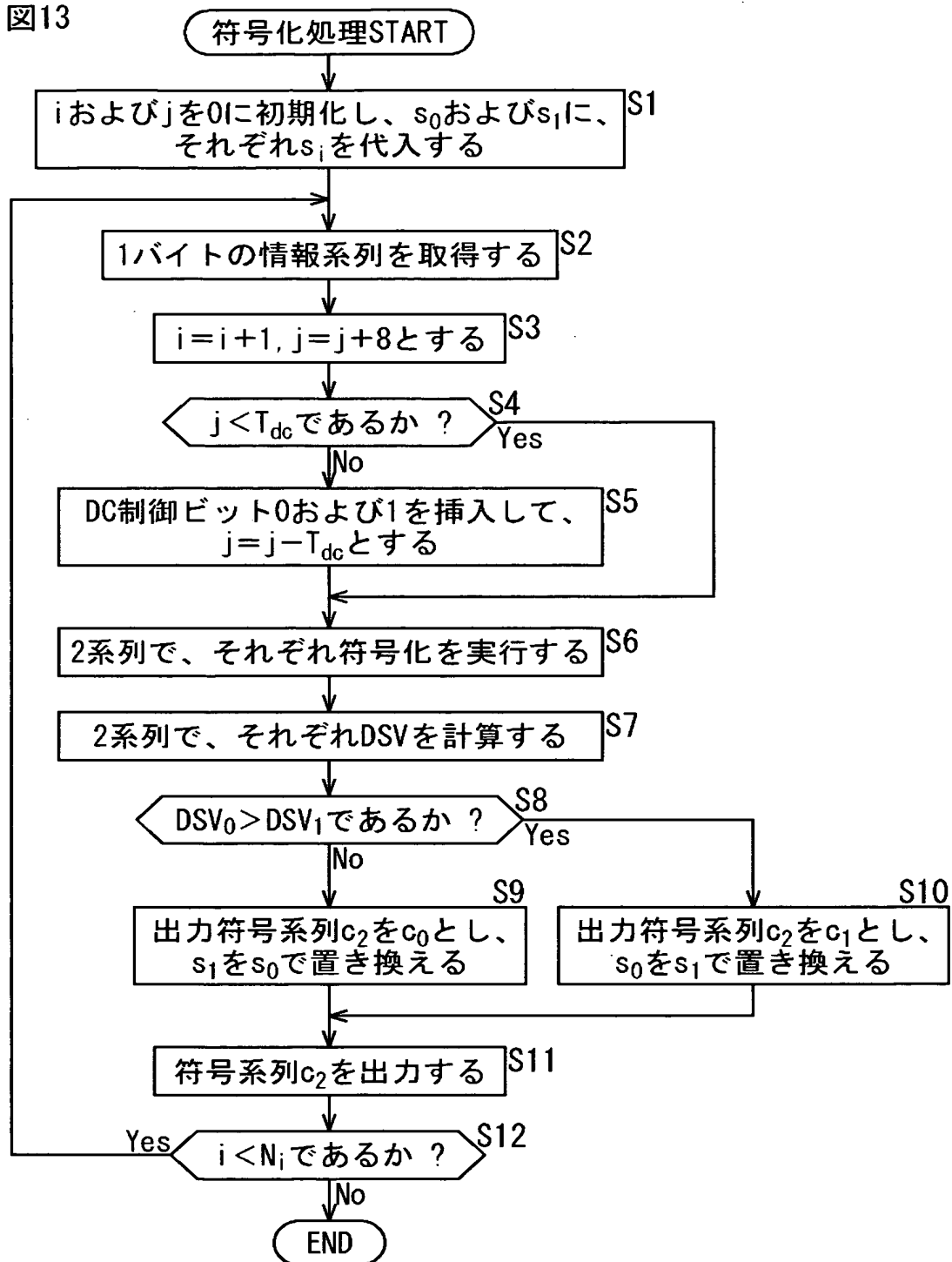
【図 12】

図12



【図13】

図13



【図 1 4】

図14

|            |     |     |     |     |
|------------|-----|-----|-----|-----|
| 情報系列 $d_0$ | 01  | 10  | 00  | 10  |
| 符号系列 $c_0$ | 100 | 101 | 000 | 010 |
| 符号状態 $s_0$ | 2   | 3   | 1   | 3   |
| 情報系列 $d_1$ | 11  | 10  | 00  | 10  |
| 符号系列 $c_1$ | 101 | 010 | 000 | 010 |
| 符号状態 $s_1$ | 4   | 3   | 1   | 3   |

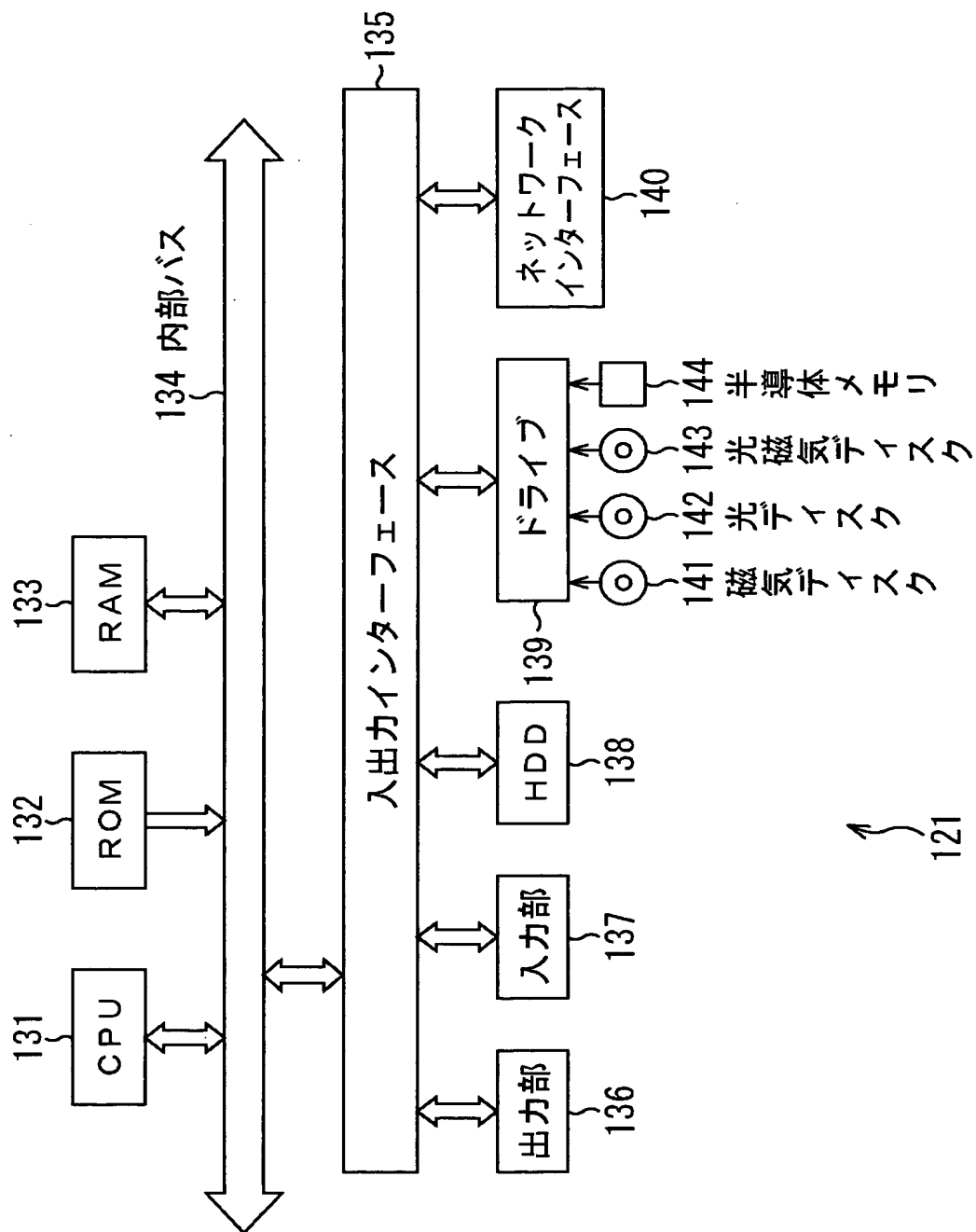
【図 1 5】

図15

|   |       |       |       |       |
|---|-------|-------|-------|-------|
| 1 | 010/3 | 010/6 | 100/3 | 100/6 |
| 2 | 010/3 | 010/6 | 000/3 | 000/6 |
| 3 | 000/1 | 000/4 | 000/5 | 000/7 |
| 4 | 010/1 | 010/4 | 010/5 | 010/7 |
| 5 | 100/1 | 100/4 | 100/5 | 100/7 |
| 6 | 001/2 | 001/3 | 001/4 | 001/6 |
| 7 | 101/2 | 101/3 | 101/4 | 101/6 |

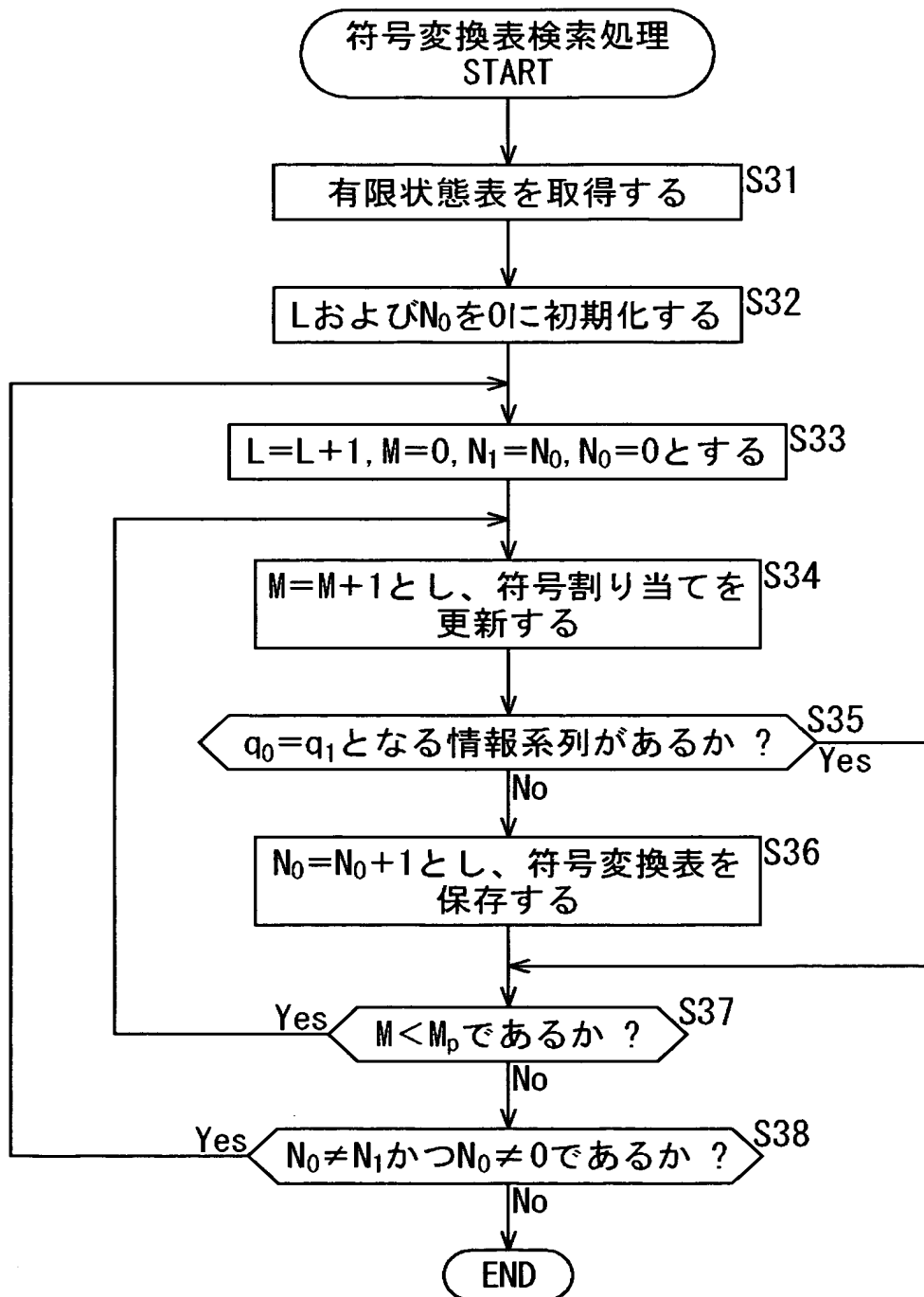
【図 16】

図16



【図 17】

図17





【図 1 8】

図18

| L<br>[ブロック] | N(L)          |              |
|-------------|---------------|--------------|
|             | 拘束長4          | 拘束長3         |
| 1           | 955, 514, 880 | 63, 700, 992 |
| 2           | 761, 384, 832 | 68, 950, 144 |
| 3           | 15, 381, 808  | 1, 845, 872  |
| 4           | 447, 800      | 83, 240      |
| 5           | 328, 272      | 65, 648      |
| 6           | 327, 680      | 65, 536      |
| 7           | 327, 680      | 65, 536      |

【図 1 9】

図19

| タイプ | 拘束長4     | 拘束長3    |
|-----|----------|---------|
| 1   | 163, 840 | 32, 768 |
| 2   | 163, 840 | 32, 768 |
| 3   | 163, 840 | 32, 768 |

【図 2 0】

図20

|   | 00    | 01    | 10    | 11    |
|---|-------|-------|-------|-------|
| 1 | 010/6 | 010/3 | 100/3 | 100/6 |
| 2 | 010/6 | 010/3 | 000/6 | 000/3 |
| 3 | 000/1 | 000/5 | 000/4 | 000/7 |
| 4 | 010/1 | 010/5 | 010/4 | 010/7 |
| 5 | 100/1 | 100/5 | 100/4 | 100/7 |
| 6 | 001/2 | 001/6 | 001/4 | 001/3 |
| 7 | 101/2 | 101/6 | 101/4 | 101/3 |

【図 2 1】

図21

|   | 00    | 01    | 10    | 11    |
|---|-------|-------|-------|-------|
| 1 | 010/3 | 010/6 | 100/6 | 100/3 |
| 2 | 010/3 | 010/6 | 000/3 | 000/6 |
| 3 | 000/5 | 000/1 | 000/7 | 000/4 |
| 4 | 010/5 | 010/1 | 010/7 | 010/4 |
| 5 | 100/5 | 100/1 | 100/7 | 100/4 |
| 6 | 001/6 | 001/2 | 001/3 | 001/4 |
| 7 | 101/6 | 101/2 | 101/3 | 101/4 |

【図 2 2】

図22

|   | 00    | 01    | 10    | 11    |
|---|-------|-------|-------|-------|
| 1 | 100/6 | 010/6 | 010/3 | 100/3 |
| 2 | 000/3 | 010/6 | 010/3 | 000/6 |
| 3 | 000/7 | 000/1 | 000/5 | 000/4 |
| 4 | 010/7 | 010/1 | 010/5 | 010/4 |
| 5 | 100/7 | 100/1 | 100/5 | 100/4 |
| 6 | 001/3 | 001/2 | 001/6 | 001/4 |
| 7 | 101/3 | 101/2 | 101/6 | 101/4 |

【図 2 3】

図23

|            |     |     |     |     |
|------------|-----|-----|-----|-----|
| 情報系列 $d_0$ | 01  | 10  | 00  | 10  |
| 符号系列 $c_0$ | 000 | 100 | 001 | 001 |
| 符号状態 $s_0$ | 1   | 6   | 6   | 3   |
| 情報系列 $d_1$ | 11  | 10  | 00  | 10  |
| 符号系列 $c_1$ | 000 | 010 | 101 | 001 |
| 符号状態 $s_1$ | 4   | 7   | 6   | 3   |

【図 2 4】

図24

| Data         | Code     |
|--------------|----------|
| 10           | 010      |
| ("0") 11     | 101      |
| ("1") 11     | 000      |
| ("0") 01     | 100      |
| ("1") 01     | 001      |
| 00. 00       | 010. 001 |
| 00. 01       | 010. 000 |
| ("0") 00. 11 | 100. 001 |
| ("1") 00. 11 | 000. 000 |
| ("0") 00. 10 | 100. 000 |
| ("1") 00. 10 | 000. 001 |

[("0")は直前の符号ビットが0]

[("1")は直前の符号ビットが1]

【図 2 5】

図25

| Data         | Code     |
|--------------|----------|
| 11           | 010      |
| ("0") 10     | 101      |
| ("1") 10     | 000      |
| ("0") 00     | 100      |
| ("1") 00     | 001      |
| 01. 01       | 010. 001 |
| 01. 00       | 010. 000 |
| ("0") 01. 10 | 100. 001 |
| ("1") 01. 10 | 000. 000 |
| ("0") 01. 11 | 100. 000 |
| ("1") 01. 11 | 000. 001 |

[("0")は直前の符号ビットが0]

[("1")は直前の符号ビットが1]

【図 2 6】

図26

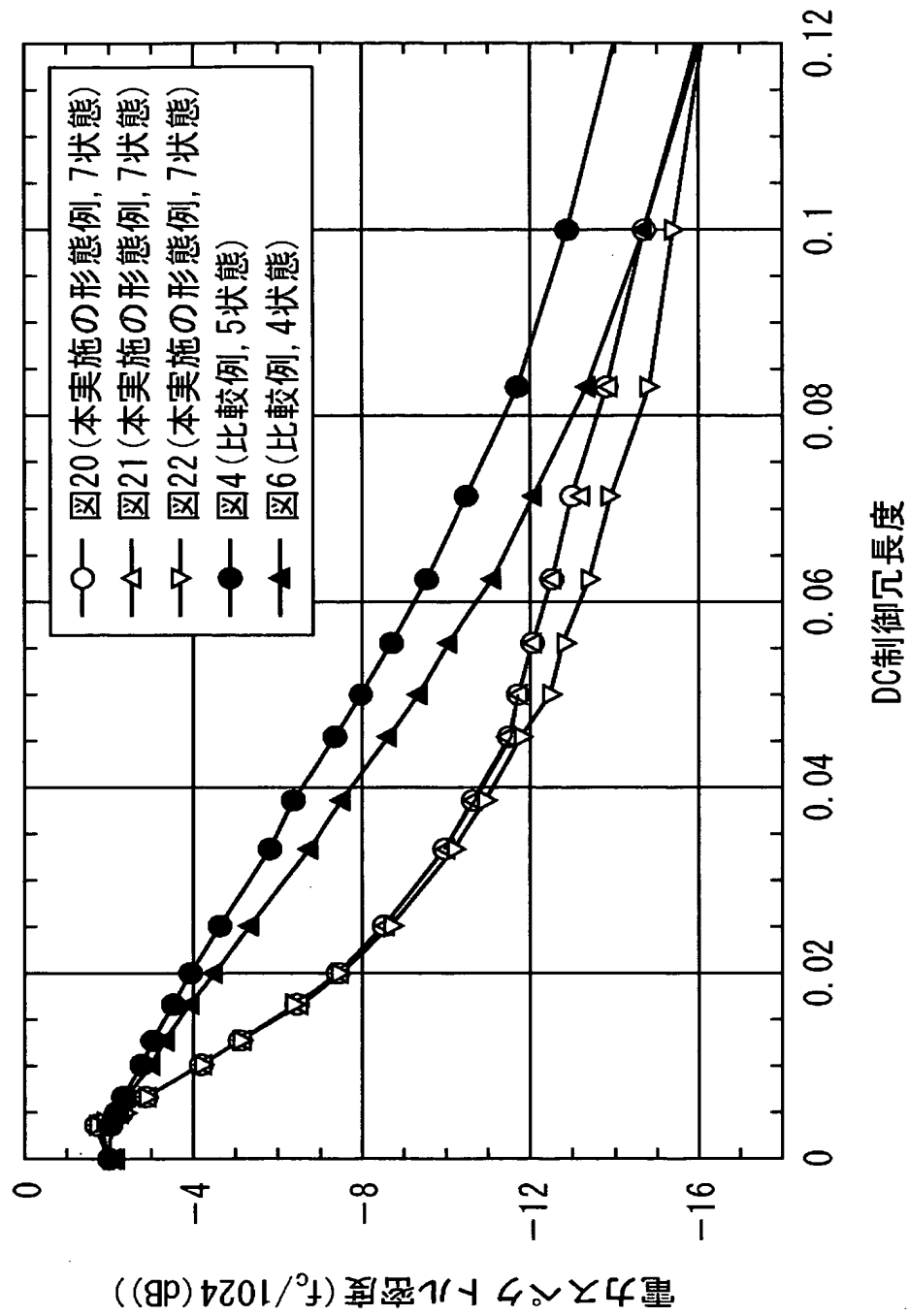
| Data       | Code    |
|------------|---------|
| 11         | 010     |
| ("0")00    | 101     |
| ("1")00    | 000     |
| ("0")10    | 100     |
| ("1")10    | 001     |
| 01.01      | 010.001 |
| 01.10      | 010.000 |
| ("0")01.00 | 100.001 |
| ("1")01.00 | 000.000 |
| ("0")01.11 | 100.000 |
| ("1")01.11 | 000.001 |

[("0")は直前の符号ビットが0]

[("1")は直前の符号ビットが1]

【図 27】

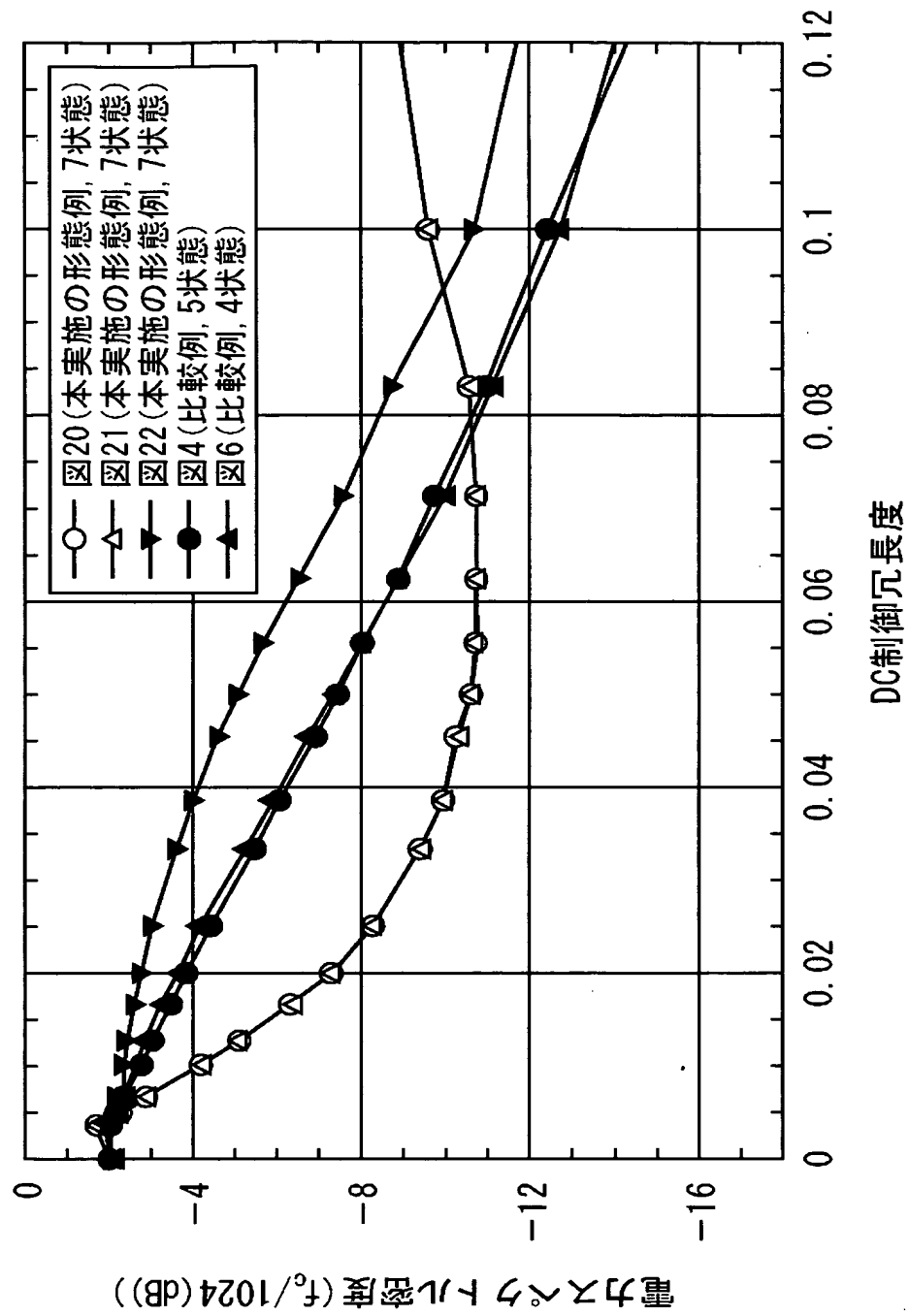
図27





【図 28】

図28



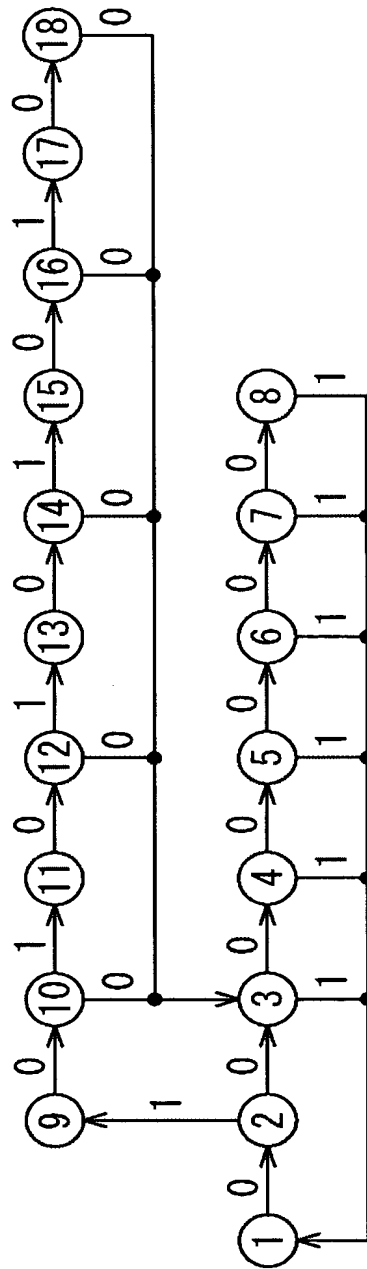
【図 2 9】

図29

| 最小ランの最大連続数 | シャノン容量  |
|------------|---------|
| 1          | 0. 6307 |
| 2          | 0. 6623 |
| 3          | 0. 6730 |
| 4          | 0. 6769 |
| 5          | 0. 6784 |
| 6          | 0. 6789 |

【図 3 0】

図30



【図 3 1】

図31

|   |       |       |       |       |
|---|-------|-------|-------|-------|
| 1 | 010/3 | 010/6 | 100/3 | 100/6 |
| 2 | 010/3 | 010/6 | 000/3 | 000/6 |
| 3 | 000/1 | 000/4 | 000/5 | 000/7 |
| 4 | 010/1 | 010/4 | 010/5 | 010/8 |
| 5 | 100/1 | 100/4 | 100/5 | 100/7 |
| 6 | 001/2 | 001/3 | 001/4 | 001/6 |
| 7 | 101/2 | 101/3 | 101/4 | 101/6 |
| 8 | 101/2 | 101/3 | 101/6 | 000/6 |

【図 3 2】

図32

| タイプ | 拘束長5     | 拘束長4         |
|-----|----------|--------------|
| 1   | 389, 120 | 4, 096 (128) |
| 2   | 389, 120 | 4, 096 (128) |
| 3   | 389, 120 | 4, 096 (256) |

【図 3 3】

図33

|   | 00    | 01    | 10    | 11    |
|---|-------|-------|-------|-------|
| 1 | 010/3 | 010/6 | 100/6 | 100/3 |
| 2 | 010/3 | 010/6 | 000/3 | 000/6 |
| 3 | 000/5 | 000/1 | 000/7 | 000/4 |
| 4 | 010/5 | 010/1 | 010/8 | 010/4 |
| 5 | 100/5 | 100/1 | 100/7 | 100/4 |
| 6 | 001/6 | 001/2 | 001/3 | 001/4 |
| 7 | 101/6 | 101/2 | 101/3 | 101/4 |
| 8 | 101/6 | 101/2 | 101/3 | 000/6 |

【図 3 4】

図34

| Data           | Code     |
|----------------|----------|
| ("010") 10. 11 | 000. 001 |

[("010")は直前の符号語が010]

【図 3 5】

図35

| 符号変換表    | 状態数 | 符号語拘束長<br>(ブロック) | 誤り伝播長<br>(ビット) |
|----------|-----|------------------|----------------|
| 図20乃至図22 | 7   | 3                | 6              |
| 図33      | 8   | 4                | 7              |

【書類名】 要約書

【要約】

【課題】 符号状態が同一となった時点までの符号系列のパリティが異なるような符号変換表を用いて符号化を行う。

【解決手段】 この符号変換表の符号語割当ては、復号時の符号語拘束長が3ブロックであり、DC制御ビットの挿入位置を情報語の第1ビット目と第2ビット目のいずれの位置に挿入しても、任意の情報系列について  $q_0 \neq q_1$  を満たす符号語割当てである。例えば、情報系列  $[1, 1, 0, 0, 0, 1, 0]$  の先頭に、仮のDC制御ビット1および0を挿入した情報系列  $d_0$  および情報系列  $d_1$  を、状態3を始点として、所定の符号変換表を用いて符号化した場合の符号状態  $s_0$  および符号状態  $s_1$  は、3ブロック目で、 $s_0 = s_1 = 6$  で同一となり、符号状態が同一となった時点までの符号系列  $c_0$  の合計の2の補数  $q_0$  は0、符号系列  $c_1$  の合計の2の補数  $q_1$  は1であり、 $q_0 \neq q_1$  が成立する。本発明は、記録再生装置、または、符号化装置に適用できる。

【選択図】 図21

特願 2 0 0 3 - 0 7 9 5 5 6

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 2 1 8 5 ]

1. 変更年月日

1 9 9 0 年 8 月 3 0 日

[変更理由]

新規登録

住 所

東京都品川区北品川 6 丁目 7 番 3 5 号

氏 名

ソニー株式会社